Privacy Games along Location Traces: *A Game-Theoretic Framework for Optimizing Location Privacy*

REZA SHOKRI, Cornell Tech, US
GEORGE THEODORAKOPOULOS, Cardiff University, UK
CARMELA TRONCOSO, IMDEA Software, Spain

The mainstream approach to protecting the privacy of mobile users in location-based services (LBSs) is to alter (e.g., perturb, hide, etc.) the users' actual locations in order to reduce the exposed sensitive information. In order to be effective, a location-privacy preserving mechanism must consider both the privacy and utility requirements of each user, as well as their overall exposed locations (which contribute to the adversary's background knowledge).

In this paper, we propose a methodology that enables the design of *optimal* user-centric location obfuscation mechanisms respecting each individual user's service quality requirements, while maximizing the expected error that the optimal adversary incurs in reconstructing the user's actual trace. A key advantage of a user-centric mechanism is that it does not depend on third party proxies or anonymizers, and so it can be directly integrated in the mobile devices that users use to access LBSs. Our methodology is based on the mutual optimization of user/adversary objectives (maximizing location privacy vs. minimizing localization error) formalized as a Stackelberg Bayesian game. This formalization makes our solution robust against *any* location inference attack, i.e., the adversary cannot decrease the user's privacy by designing a better inference algorithm as long as the obfuscation mechanism is designed according to our privacy games.

We develop two linear programs that solve the location privacy game and output the optimal obfuscation strategy and its corresponding optimal inference attack. These linear programs are used to design location privacy-preserving mechanisms that consider the correlation between past, current and future locations of the user, thus can be tuned to protect different privacy objectives along the user's location trace. We illustrate the efficacy of the optimal location privacy-preserving mechanisms obtained with our approach against real location traces, showing their performance in protecting users' different location privacy objectives.

CCS Concepts: •**Security and privacy** → **Privacy-preserving protocols;** *Mobile and wireless security;* •**Information systems** → **Location based services;**•**Networks** → **Network privacy and anonymity;** •**Mathematics of computing** → *Linear programming;*

Additional Key Words and Phrases: Location Privacy, Utility, Optimization, Game Theory

ACM Reference Format:
Reza Shokri, George Theodorakopoulos, and Carmela Troncoso. 2016. Privacy Games along Location Traces: A Game-Theoretic Framework for Optimizing Location Privacy. *ACM Trans. Priv. Secur.* 1, 1, Article 1 (January 2015), 30 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The widespread use of smart mobile devices with continuous connection to the Internet has fostered the development of a variety of successful location-based services (LBSs). Even though LBSs can be very useful, these benefits come at a cost for users' privacy. The whereabouts of users disclose aspects of their private lives that are not apparent

Authors' addresses: Reza Shokri, Cornell Tech, #302, 111 Eight Ave. New York, US; George Theodorakopoulos, School of Computer Science and Informatics, Cardiff University, Cardiff CF24 3AA, UK; Carmela Troncoso, IMDEA Software Institute, 28223-Pozuelo de Alarcón, Madrid, Spain.

DOI: http://dx.doi.org/10.1145/0000000.0000000

at first, but can be inferred from the revealed location data [Freudiger et al. 2012; Golle and Partridge 2009; Krumm 2007].

A large body of research has focused on developing location-privacy protection mechanisms (LPPMs) that allow users to make use of LBSs while limiting the amount of disclosed sensitive information [Beresford and Stajano 2003; Chow and Golle 2009; Freudiger et al. 2009; Gedik and Liu 2005; Hoh et al. 2007; Kalnis et al. 2007; Meyerowitz and Roy Choudhury 2009]. These protection mechanisms are based on increasing the uncertainty of the adversary about a user's true whereabouts by hiding locations from the LBS, or sending perturbed or fake locations. However, more often than not the evaluation of these designs disregards that the adversary might have some prior knowledge about users' movements as well as about the algorithm implemented by the LPPM. It has been shown [Shokri et al. 2011a] that such information allows a *strategic adversary* to reduce her uncertainty on the user's true location, and hence prior evaluations overestimate the location privacy offered by a given protection system.

Furthermore, previous work usually targets protection against localization attacks [Shokri et al. 2011a; Shokri et al. 2011b], i.e., to protect the users' whereabouts when disclosing an obfuscated location to the LBS. Protecting the user's current location is intricately bound with protecting her past and future locations. Different LBSs require the user location to be updated at different rates. Some require frequent updates, such as applications for obtaining local traffic information – e.g., Beat the Traffic, INRIX Traffic Maps, Routes & Alerts; or finding nearby points-of-interest, local events, or nearby friends – e.g., Google Maps. Others can function perfectly well with just a single location, such as Foursquare, Google Latitude / Google+ Local, or, in general, location check-in and location-tagging services that require only sporadic location exposures. The frequency of location exposures has a severe impact on the privacy protection offered by an LPPM since locations exposed in quick succession are highly correlated, leaking a lot of information that allows the adversary to reduce her uncertainty on the user's immediate past or future whereabouts.

In this work we propose an LPPM design methodology that explicitly accounts for a strategic adversary, allowing for a more accurate estimation of the privacy protection that can be achieved by these LPPMs. The design methodology takes *location correlation* into account to effectively protect a user's location privacy along her trajectory, e.g., it provides protection against inference attacks that aim at reconstructing the user's location in the past or predicting her location in the future based on what she shares at any moment. We focus on *user-centric* LPPMs in which privacy decisions are made locally by the user, rather than by a third party that acts as an anonymizer [Gruteser and Grunwald 2003]. These LPPMs require no changes in the infrastructure nor any cooperation from other users, from third parties, or from the LBS; hence, they can be directly integrated into the mobile devices.

The goal of our methodology is to allow system designers to find the *optimal LPPM* against a strategic adversary who, knowing each user's LBS access pattern, observation history, and the LPPM's obfuscation algorithm, employs the theoretically strongest attack to infer users' whereabouts. The challenge is to design such an optimal protection mechanism when the inference attack, that the strategic adversary will choose depending on the mechanism being designed, is a priori unknown to the designer. To overcome this obstacle, instead of making assumptions about the adversary's inference algorithm (i.e., instead of assuming limits in his power), our approach anticipates the optimal attack. Additionally, our methodology constrains the search space to LPPMs that obfuscate locations in such a way that the quality of the LBS response is not degraded below a threshold imposed by the user, hence guaranteeing the required service quality for the user. We assume that this user-specified service quality constraint is also known to the adversary.

We formalize the problem of finding the optimal LPPM anticipating the optimal inference attack as an instance of a zero-sum Bayesian Stackelberg *game*. In this game, a leader and a follower interact strategically, with each one's gain being the loss of the other. The leader decides on her strategy knowing that it will be observed by the follower, who will optimize his choice based on this observation. In our scenario the user is the leader and the adversary is the follower. Then, the game precisely models that the adversary knows the user's choice of protection mechanism and will use that knowledge to improve his attack's effectiveness. We further extend the classic formulation of a Stackelberg game with an extra constraint to ensure that the service quality is satisfactory for the user. This enables us to find the optimal point in the tradeoff curve between privacy and service quality that satisfies users' requirements. We prove that the solution to our location privacy games can be obtained using linear programs that must be solved by the user along the trace of her movements, since, as she reveals more of her location over time, the adversary's knowledge and observation history evolves. Ours is, to the best of our knowledge, the first analytical framework that allows us to integrate adversarial knowledge in the design methodology of optimal user-centric privacy protection mechanisms.

We apply our methodology to design LPPMs for various scenarios in which we aim at protecting not just the current user location, but also at protecting past locations (i.e. the LPPM's current obfuscation is chosen so as not to compromise the privacy of past locations), future locations (i.e. the current obfuscation should be compatible with potential future locations where the user might go next), and transitions between locations (i.e. obfuscations of successive locations should be chosen jointly).

We evaluate the effectiveness of the designed LPPMs using real location traces, showing that for a given user's LBS access pattern and service-quality threshold, our game-theoretic approach enables us to simultaneously find the optimal LPPM and the optimal attack against it. We confirm that there is a trade-off between the maximum achievable privacy and the service quality but, once a certain privacy level is reached, loosening the quality requirements does not necessarily result in a privacy gain. We also find that the location-privacy advantage of the optimal LPPM over a suboptimal LPPM is larger when the quality requirement is tighter: The intuition is that, when the quality requirement is loose, both LPPMs are allowed to add so much noise that the attacker's observation is very uninformative about the user's true location. In contrast, when the quality requirement is tight, the optimal LPPM makes a better allocation of the limited noise it is allowed to inject.

In summary, our proposal for designing LPPMs showcases four contributions:

(1) Our LPPMs assume that the adversary is strategic, because, as Shannon's maxim states "One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them."
(2) Our LPPMs are user-centric, needing no changes in the infrastructure, nor any trusted third-parties, because we believe that would facilitate their deployment.
(3) Our LPPMs take into account that successively visited locations are correlated, which dramatically changes the way they should be protected.
(4) Our LPPMs are optimal by design in the scenarios that they are designed for, hence they offer the best possible privacy protection against the best possible attack.

We motivate and state the problem in the next two sections, and formalize the problem as a Bayesian Stackelberg game between user and adversary in Section 4. We show how to design optimal protection mechanisms in different scenarios in Section 5, and evaluate our method in Section 6. We discuss the related work in Section 7.

2. MOTIVATION: LOCATION PRIVACY AGAINST STRATEGIC ADVERSARIES

This section delves into the need to consider, at the time of designing an LPPM, strategic adversaries that use their knowledge of user behavior, LPPM operation, and correlation between successively visited locations, since the use of such knowledge has great impact on the maximum level of privacy achievable.

2.1. Strategic adversaries use prior knowledge on users' behavior and LPPM design to reduce their uncertainty

Consider a user who issues a location-based query from location (2,2), ★, in the area depicted in Figure 1. Instead of revealing her true location, she uses an LPPM that outputs a 3x3 square, ★, centered on her true location. The hope is that this mechanism masks her location, making any location within the exposed square equally probable in the eyes of the attacker.

There are two problems with this naive approach. First, *if the adversary has background information about the user's mobility*, the probability of the user being in any location is not uniform anymore. For example, the adversary may have access to topographic information about the area in which the user moves, which changes the probability of the user visiting locations: The user is much more likely to be at the shore of a lake than in its middle. This probability also changes if the attacker has access to behavioral information about the user, e.g. her favorite locations reported in social networks, or demographic information such as her age, which lead to likely locations of that particular demographic group. Second, *if the adversary knows how the mechanism works*, the probability of the user being in any location may also change. In the example above, such knowledge would completely break the user's privacy. Upon observing the reported square, the attacker would always correctly infer that the user is located at (2,2), the center of the square.

These examples show that not considering a strategic adversary with access to background knowledge leads to an overestimation of the privacy achievable by an LPPM.

2.2. Strategic adversaries use correlation to reduce their uncertainty

Users' movements are not isolated discrete events. Rather, users follow a *trajectory* to go from one place to another. Along this trajectory, users may query a location-based service, continuously in time or only at selected spots, to obtain useful information concerning the surroundings or the arrival point. However, even if not all points in the trajectory are exposed to the service provider, the correlation between consecutive positions implies that inferring just one of them reveals information about past and future ones. For instance, spatio-temporal constraints derived from maximum user velocity may reveal with high probability the route followed by a user between two successive location exposures.

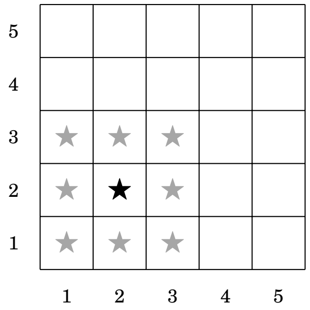
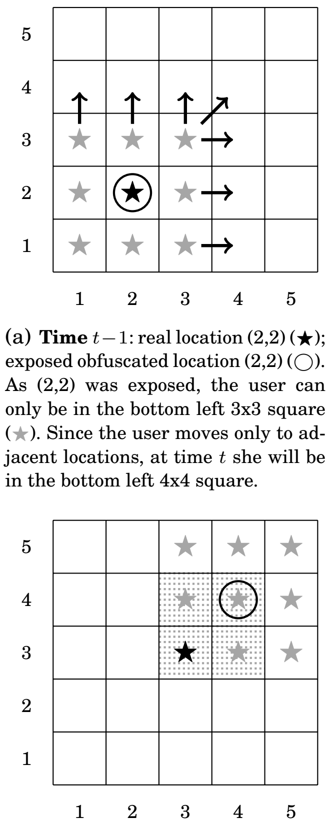


Fig. 1. A user's real location (★) and obfuscated exposed location (★). In principle, the attacker has 1 chance in 9 to accurately locate the user. However, if the attacker knows that row 1 is a lake, or that the protection mechanism always produces a 3x3 square centered on the real location, then the user's privacy is lower than naively expected.

The correlation between successively shared locations depends on two factors: randomness of user mobility patterns and LBS access frequency. The former relates to how predictable a user's future location is given her current location. The latter defines the rate at which the LBS provider can sample the user's trajectory. These two factors have opposite effects on correlation: High randomness decreases correlation between successively exposed locations, since the current position contains less information about past and future events than when movements are deterministic. In contrast, high LBS access frequency increases correlation: The user has little time to move between two LBS accesses and exposed locations are nearer to each other than when access frequency is low. We now show, through a toy example, how correlation between successive locations can decrease the uncertainty of the adversary about past and/or future locations of the user.

Consider the example in Figure 2 where the LPPM outputs a location chosen arbitrarily from the 3x3 square centered on (x, y), $\{(x+i, y+j), i, j \in \{-1, 0, 1\}\}$, and where in addition the user moves at most one location per time unit. At time $t-1$, shown in Figure 2(a), the user accesses the LBS from location (2,2), ★, and the LPPM reports obfuscated location (2,2), ○. Given the reported location and the LPPM mechanism, the adversary can infer that the user could only have been in the bottom-left 3x3 square of locations, ☆. Moreover, since the user moves at most one location per time unit, the adversary knows that at time t she will be somewhere inside the bottom-left 4x4 square (as shown by the arrows in Figure 2(a)).

At time t, shown in Figure 2(b), the user accesses again the LBS from location (3,3), ★, reporting obfuscated location (4,4), ○. Naively, as in Section 2.1, the probability of the adversary correctly guessing her real location is 1/9 (any location in the 3x3 square surrounding location (4,4), ☆). However, recall that the adversary knows from the observation at $t-1$ that at time t the user can only be in the bottom-left 4x4 square. Intersecting this knowledge with the current observation the adversary can deduce that the user is in the darkened 2x2 square in Figure 2(b). Therefore, the probability of a correct guess is 1/4, more than twice as much as the naively expected 1/9. This example highlights that *designing LPPMs disregarding correlation may reduce the privacy of the current location*. We formally describe this scenario as Problem 1 in Section 5, and we also show how to optimally choose obfuscated exposed locations to avoid this situation.



(a) **Time** $t-1$: real location (2,2) (★); exposed obfuscated location (2,2) (○). As (2,2) was exposed, the user can only be in the bottom left 3x3 square (☆). Since the user moves only to adjacent locations, at time t she will be in the bottom left 4x4 square.

(b) **Time** t: real location (3,3) (★); exposed obfuscated location (4,4) (○). As (4,4) was exposed, the user can only be in the top right 3x3 square (☆). However, the correlation with the previous disclosure implies that the user can only be in the dotted 2x2 square.

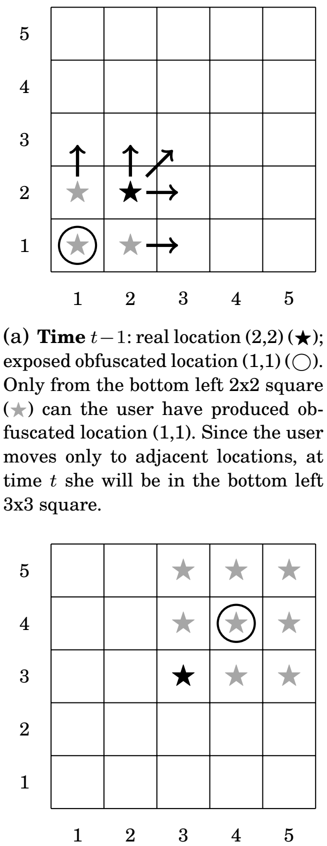
Fig. 2. Correlation of user's locations between time $t-1$ and time t, reduces attacker's uncertainty about the user's current (time t) location.

Now consider the example in Figure 3, where at time $t-1$ (Figure 3(a)) the user is at location (2,2), ★, but the LPPM reports (1,1), ○, instead of reporting (2,2); and at time t (Figure 3(b)) the user is at (3,3), ★, and the LPPM reports (4,4), ○. In this case, the correlation stemming from the user's one-location-per-time-unit movement pattern compromises the real locations at *both* $t-1$ *and* t: The only two-step trajectory that is compatible with the successive exposure of obfuscated locations (1,1) and (4,4) given the LPPM operation is that the user accessed the LBS from (2,2) followed by (3,3). Strikingly, *designing LPPMs disregarding correlation may retroactively compromise the privacy of past locations*: the user was safe until the obfuscated location at time t was reported. In Section 5, Problem 2, we show how to choose obfuscated exposed locations at time t so as not to compromise neither the previous nor the current location.

The example just discussed also illustrates that the choice of obfuscated locations can affect *future* privacy. Consider again the user's predicament at time t: She reported (1,1) at time $t-1$ and she is now at (3,3) trying to choose an appropriate obfuscated location. Revealing (4,4) is not an option since it would reveal her real location at both $t-1$ and t. Hence, the unfortunate choice at $t-1$ reduces her possible choices at t. In other words, *future privacy may be proactively compromised by current choices*. This scenario is handled as Problem 3 in Section 5, where we offer a mechanism to choose obfuscated exposed locations so as not to compromise future locations.

Finally, taking into account correlation between successively visited places is important because *transitions* between locations can also be sensitive, even when the individual locations on their own might not be. For instance, visiting the bank to make a big withdrawal, and visiting a government official in charge of land development licensing may not be very sensitive if considered separately, but visiting the official immediately after the bank may be much more sensitive. Another sensitive issue with transitions between locations is that they reveal the direction of travel. For instance, the adversary may learn whether the user enters or exits a building, e.g., a hospital.

The conclusion from these examples is that it is important to take into account the correlation between exposed locations when designing LPPMs for frequently queried location-based services, in order to protect all past, current and future locations. In the following we introduce our mechanism that allows to find the optimal way to expose obfuscated locations, i.e., maximizing privacy in the best possible way against a strategic adversary.



(a) **Time** $t-1$: real location (2,2) (★); exposed obfuscated location (1,1) (◯). Only from the bottom left 2x2 square (★) can the user have produced obfuscated location (1,1). Since the user moves only to adjacent locations, at time t she will be in the bottom left 3x3 square.

(b) **Time** t: real location (3,3) (★); exposed obfuscated location (4,4) (◯). Only from the top right 3x3 square (★) can the user have produced obfuscated location (4,4). However, the attacker correlates the previous disclosure with the current one and concludes that at $t-1$ the user could only be at location (2,2), and at t she can only be at (3,3).

Fig. 3. Correlation of user's locations between time $t-1$ and time t, reduces attacker's uncertainty about both the user's current (time t) location and her past (time $t-1$) location.

3. PROBLEM STATEMENT

In this section, we first explain our probabilistic framework and our assumptions about the protection mechanisms and adversary model. We conclude by sketching the problem that we solve in this paper. In Table I, we summarize the notations introduced throughout the section.

3.1. User and Adversary

We consider a scenario in which users move in an area partitioned into M discrete regions $\mathcal{R} = \{r_1, r_2, \cdots, r_M\}$. We also assume that time is discrete and it is partitioned into different time periods (e.g., morning, afternoon). An *event* $\langle r, t \rangle$ denotes that the user is at location $r \in \mathcal{R}$ at time $t \in \mathcal{T}$. Slightly abusing notation, time-subscripted variable r_t will denote the user's location at time t. Typical values from $\mathcal{R}$ are r, r_i, r_j, whereas typical time-subscripted locations are r_t, r_{t-1}, r_{t+1}. As the user moves, she connects to an LBS with which she shares her current location in order to obtain a service.

The adversary is the LBS to which the user connects, or any entity that can eavesdrop on the user-LBS communications. The adversary is passive and curious, and his aim is to discover the location of the user at the query time. He observes the locations sent by the user (protected by the privacy mechanism, as described in the next subsection), and also has some external information about the user's mobility. For simplicity we assume the user's mobility is encoded as a Markov chain, although that is not mandatory. Any other model is possible, as long as it allows us to compute probabilities of the user visiting various (sequences of) locations.

Let $\psi_t(r)$ be the probability, from the point of view of the adversary, that the user accesses the LBS at time t from location r. So, $\sum_{r \in \mathcal{R}} \psi_t(r) = 1$ for any time instant t. We note that this quantity is time-dependent (i.e., users may have different access patterns in the morning than in the afternoon). In addition, $\psi_t(r)$ can depend on previous LBS accesses that the user has made and the adversary has observed; this dependence on previous accesses will be shown explicitly when needed. Overall, ψ_t can be computed by the adversary from the combination of the user's general mobility and her previous LBS accesses. We call ψ the *user's profile* as it reflects the behavior of the user in accessing the LBS over time, as seen from the point of view of the attacker.

Note, crucially, that $\psi_t(r)$ is the adversary's *prior* information about the user's location at time t, *before* observing anything at time t.

3.2. Location-Privacy Protection Mechanism

As users want to preserve their location privacy when they use the LBS, they employ a local and user-centric LPPM that transforms each true location r into a *pseudolocation* (or *obfuscated location*) $r' \in \mathcal{R}'$, which is then sent to the LBS instead of the actual location. For simplicity, we set $\mathcal{R}' = \mathcal{R}$, though in the most general case, $\mathcal{R}'$ is the powerset of $\mathcal{R}$. Indeed, Problem 0 (Section 5) has been extended in this direction [Herrmann et al. 2013]. The corresponding events $\langle r', t \rangle$ are termed *pseudoevents* or *observed events*.

The transformation from r to r' at time t happens probabilistically according to the location obfuscation function implemented by the LPPM: a probability distribution $f_t(r'|r)$, which can be seen as a matrix whose rows are indexed by the locations $r \in \mathcal{R}$ and whose columns are indexed by the pseudolocations $r' \in \mathcal{R}'$. For a given location r the function $f_t(r'|r)$ defines the probability with which the LPPM selects pseudolocation r' as output.

Table I. Summary of notations

Symbol	**Meaning**
r, $\mathcal{R}$, M	Actual location of the user, set of possible locations, number of locations
$\psi_t(r)$	User's profile: the probability, according to the attacker, of the user being at location r when accessing the LBS at time t.
$\langle r, t\rangle$ or r_t	Actual event: the user is at location r at time t
r', $\mathcal{R}'$	User's pseudolocation as the output of the LPPM, and the set of possible pseudolocations
r'_t	observed pseudolocation r' of the user at time t
$f_t(r' \mid r)$	Location obfuscation function implemented by the LPPM: Probability of replacing r with r' at time t
$d_q(r', r)$	Incurred service-quality loss by the user if LPPM replaces location r with pseudolocation r'
$Q_{loss}(\psi_t, f_t, d_q)$	Expected quality loss of an LPPM at time t, given profile ψ_t and the location obfuscation function f_t
$Q_{loss}^{\max}$	Maximum tolerable service quality loss
$\hat{r}$	Adversary's estimate of the user's location
$h_t(\hat{r} \mid r')$	Adversary's inference attack function: Probability of estimating $\hat{r}$ as the user's actual location, if r' is observed from the user at time t
$d_p(\hat{r}, r)$	Distance between locations $\hat{r}$ and r: Privacy of the user at location r if adversary's estimate is $\hat{r}$
$Privacy(\psi_t, f_t, h_t, d_p)$	Expected location privacy of the user at time t, given profile ψ_t using protection f_t against attack h_t

3.3. Service Quality Metric

In the aforementioned setting, the LBS response quality depends on the pseudolocation output by the LPPM and not on the user's actual location. The distortion introduced in the observed pseudolocations determines the quality of service that the user experiences. We model the loss of service quality due to obfuscation using a distance function $d_q(r', r)$.[1] This function quantifies the dissimilarity between location r and pseudolocation r'. Its value for any pair (r, r') depends on how the LBS under consideration responds to obfuscated locations, as studied e.g. in [Micinski et al. 2013], and also on the user's specific service-quality expectations. In many applications, the service quality can be considered inversely proportional to the physical distance between r and r'. For example, applications that find nearby points of interest could give very different responses to r and to r' even if they are only a couple of kilometers apart. In contrast, there exist LBSs in which the service quality depends on other criteria, such as whether r' is within a region of interest. For a weather forecast application, for instance, any pseudolocation r' in the same city as the actual location r would result in a high quality LBS response. In general, d_q can model a broad range of service quality loss functions, and it is considered as an input to our framework.

The *expected quality loss* Q_{loss} due to an LPPM f_t is computed as an average of $d_q(r', r)$ over all r and r':

$$Q_{loss}(\psi_t, f_t, d_q) = \sum_{r,r'} \psi_t(r) \cdot f_t(r'|r) \cdot d_q(r', r). \qquad (1)$$

[1] The quality loss function could also be time dependent, e.g., users could be more concerned about quality during working hours than during their free time.

We assume that users impose a maximum tolerable service quality loss, $Q_{loss}^{\max}$, so as to prevent the obfuscation function from making the service useless for them. Formally,

$$Q_{loss}(\psi_t, f_t, d_q) \leq Q_{loss}^{\max}. \tag{2}$$

This constrains the LPPM obfuscation function $f_t(r'|r)$, which must not output pseudolocations that, on average, result in a lower quality. We note that the influence of the threshold $Q_{loss}^{\max}$ on the LPPM depends on the function d_q, hence it is also dependent on the type of the LBS the user is querying. In the case of an LBS that finds nearby points of interest, where d_q is proportional to the physical distance between r and r', enforcing the quality threshold could result in ensuring a maximum allowed distance between these two locations. For the weather application, enforcing the quality threshold could result in setting region boundaries within which locations lead to the same forecast. For other location-based applications, the function d_q and the threshold $Q_{loss}^{\max}$ can be defined in the same vein.

3.4. Location Privacy Metric

The adversary's goal is to infer the user's true location r_t after observing the LPPM's output r'_t at time t. The adversary uses his knowledge of the user's profile ψ_t to run an inference attack on the observed location r'_t in order to output estimations $\hat{r}_t$ of the user's actual location. Formally, the attack result can be described as a probability distribution function $h_t(\hat{r}_t|r'_t)$, which denotes the probability, according to the adversary, that the user's true location at time t is $\hat{r}_t$.

We follow the definition by Shokri *et al.* [2011b] and quantify the user's location privacy as the adversary's expected error in his inference attack, i.e., the expected distortion $d_p(\hat{r}, r)$ between the estimated location $\hat{r}$ and the true location r. We compute the expectation over all r, r', and $\hat{r}$:

$$Privacy(\psi_t, f_t, h_t, d_p) = \sum_{\hat{r}, r', r} \psi_t(r) \cdot f_t(r'|r) \cdot h_t(\hat{r}|r') \cdot d_p(\hat{r}, r) \tag{3}$$

The distortion function $d_p(.)$ quantifies the privacy that still persists despite the inference attack. This level of privacy depends on the locations' semantics and also on the privacy requirements of the user (e.g., users might consider locations inside a hospital more sensitive than other places), and $d_p(.)$ must be defined accordingly. For instance, if the user wants to hide just her exact current location (as opposed to hiding a large area around her location), the appropriate distortion function could be the Discrete Metric between the estimated location $\hat{r}$ and the actual location r:

$$d_p(\hat{r}, r) = \begin{cases} 0, & \text{if } \hat{r} = r \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

Substituting this $d_p(.)$ into (3) and performing the summation, we see that the term with $\hat{r} = r$ disappears, and all terms with $\hat{r} \neq r$ survive with $d_p(\hat{r}, r) = 1$. In other words, the resulting sum is the total probability attributed to the wrong estimations of r, i.e., the probability of error in the estimate.

Alternatively, the user's privacy might depend on the *physical* distance between the estimated and actual locations. In that case, the distortion function can be equal to the Squared Euclidean distance between these locations:

$$d_p(\hat{r}, r) = (\hat{r} - r)^2 \tag{5}$$

In general, d_p reflects the sensitivity of the user when r is estimated as $\hat{r}$. This sensitivity can be due to any semantic relation between r and $\hat{r}$. We assume d_p is an input to our framework.

3.5. Problem Statement

Having introduced all the components of our framework, we are ready to state precisely the problem that we solve. Given:

(a) a maximum tolerable service-quality loss $Q_{loss}^{\max}$ imposed by the user as a bound for Q_{loss}, computed using the quality function d_q, and
(b) a user profile ψ_t, computed from the user's mobility pattern and from her previously observed locations,

we find the LPPM obfuscation function f_t that maximizes the user's location privacy as defined in (3). The solution must consider that the adversary

(a) observes the LPPM's output r', and
(b) is aware of the LPPM's internal algorithm f_t, and the user's profile ψ_t.

The adversary implements the *optimal* attack h_t that estimates the true location of the user with the least distortion as measured by d_p.

4. LOCATION PRIVACY GAMES

We formulate the problem of designing LPPMs that are optimal against the strongest strategic adversary as a game. In fact, the problem of finding an LPPM that offers optimal location privacy given the user's profile, at a given time instant, is an instance of a zero-sum Bayesian Stackelberg game. In a Stackelberg game the *leader*, in our case the user, plays first by choosing an LPPM and committing to it by running it on her actual location. The *follower*, in our case the adversary, plays next by estimating the user's location, knowing the LPPM that the user has committed to. It is a Bayesian game because the adversary has incomplete information about the user's true location, and plays according to his hypothesis about this location. It is also an instance of a zero-sum game, as the adversary's gain (or loss) is exactly balanced by the loss (or gain) of the user: the information gained (lost) by the adversary is the location privacy lost (gained) by the user, according to the location privacy metric (3). We now proceed to define the steps of the game adapted to our problem:

Step 0. At time t, Nature uses probability distribution ψ_t (reflecting the probabilistic model for the mobility and the previously observed locations of the user) to select a location $r \in \mathcal{R}$ for the user, from which the user accesses the LBS.

Step 1. Given the user's location r, the LPPM uses $f_t(r'|r)$ to select a pseudolocation $r' \in \mathcal{R}'$, subject to f_t complying with the service quality constraint (2).

Step 2. Having observed r', the adversary selects an estimated location $\hat{r} \sim h_t(\hat{r}|r'), \hat{r} \in \mathcal{R}$. The adversary knows the LPPM's probability distribution $f_t(r'|r)$; he also knows the user's profile ψ_t, but of course not the true location r.

Final Step. The adversary pays an amount $d_p(\hat{r}, r)$ to the user. This amount represents the adversary's error (equivalently, the location privacy of the user).

The above description is common knowledge to both the adversary and the user. They both aim to maximize their payoff, i.e. the adversary tries to minimize the expected amount that he will pay, while the user tries to maximize it. Next, we describe an optimization problem that formalizes the objectives of the user and of the adversary. We construct two linear programs that, with inputs ψ_t, d_p and d_q, compute the user's optimal choice of protection mechanism f_t as well as the adversary's optimal choice of inference attack h_t. We emphasize that, when we use the terms "optimal f_t"

and "optimal h_t" in this paper, we mean that they are optimal *against each other*, i.e. they form a game-theoretic equilibrium. This is the kind of solution one looks for in a game.

4.1. Optimal Strategy for the User

The adversary observes the pseudolocation r' output by the LPPM, he knows the function $f_t(r'|r)$ implemented by the LPPM, and he also knows the user's profile $\psi_t(.)$. Thus, he can form the posterior distribution

$$\Pr(r|r') = \frac{\Pr(r, r')}{\Pr(r')} = \frac{f_t(r'|r)\psi_t(r)}{\sum_r f_t(r'|r)\psi_t(r)} \tag{6}$$

on the true location r of the user, conditional on the observation r'. The adversary's objective is then to choose $\hat{r}$ to minimize the user's conditional expected privacy, where the expectation is taken under $\Pr(r|r')$. The user's conditional expected privacy for an arbitrary $\hat{r}$ is

$$\sum_r \Pr(r|r') d_p(\hat{r}, r), \tag{7}$$

and for the minimizing $\hat{r}$ it is

$$\min_{\hat{r}} \sum_r \Pr(r|r') d_p(\hat{r}, r). \tag{8}$$

If there are multiple values of $\hat{r}$ that satisfy (8), then the adversary may randomize arbitrarily among them, including selecting one of them with probability 1. The probability with which $\hat{r}$ is chosen in this randomization is $h_t(\hat{r}|r')$. Of course, $h_t(\hat{r}|r')$ can be nonzero only for minimizing values of $\hat{r}$; for all other values, $h_t(\hat{r}|r')$ will be zero. When randomizing, (8) is rewritten as

$$\sum_{r,\hat{r}} \Pr(r|r') h_t(\hat{r}|r') d_p(\hat{r}, r). \tag{9}$$

Note that if there is only one value of $\hat{r}$ satisfying (8), then this value is selected with probability 1 in the randomization, whereas all other values are selected with probability 0, so (9) reduces to (8). In this sense, (9) is a generalization of (8), but it should be noted that both expressions compute the same conditional expected privacy.

We see that for a given r', the user's conditional privacy is given by (8). The probability that r' is output by the LPPM is $\Pr(r') = \sum_r f_t(r'|r)\psi_t(r)$. Hence, the user's *unconditional* expected privacy (averaged over all r') is

$$\sum_{r'} \Pr(r') \min_{\hat{r}} \sum_r \Pr(r|r') d_p(\hat{r}, r) = \sum_{r'} \min_{\hat{r}} \sum_r \psi_t(r) f_t(r'|r) d_p(\hat{r}, r). \tag{10}$$

To facilitate the computations, we define

$$x_{r'} \triangleq \min_{\hat{r}} \sum_r \psi_t(r) f_t(r'|r) d_p(\hat{r}, r). \tag{11}$$

Incorporating $x_{r'}$ into (10), we rewrite the unconditional expected privacy as

$$\sum_{r'} x_{r'}, \tag{12}$$

which the user aims to maximize by choosing the optimal $f_t(r'|r)$. The minimum operator makes the problem non-linear, which is undesirable, but (11) can be transformed to a series of linear constraints:

$$x_{r'} \leq \sum_{r} \psi_t(r) f_t(r'|r) d_p(\hat{r}, r),\ \forall \hat{r}. \tag{13}$$

It turns out that maximizing (12) under (11) is equivalent to maximizing (12) under (13) [Dasgupta et al. 2008, Ch. 7, p. 224].

We construct the linear program for the user from (12) and (13). Note that variable $x_{r'}$ is a *decision* variable in the linear program, i.e. it is among the quantities chosen by the solver. This might appear counterintuitive, as $x_{r'}$ is defined in (11) as a function of $f_t(.)$, rather than as an independent variable that can be freely selected. But, because of the transformation, it is always guaranteed that (11) will hold.

The linear program for the user is the following: Choose $f_t(r'|r), x_{r'}, \forall r, r'$ in order to

$$\textbf{Maximize} \ \sum_{r'} x_{r'} \tag{14a}$$

$$\textbf{subject to} \ x_{r'} \leq \sum_{r} \psi_t(r) f_t(r'|r) d_p(\hat{r}, r),\ \forall \hat{r}, r' \tag{14b}$$

$$\sum_{r} \psi_t(r) \sum_{r'} f_t(r'|r) d_q(r', r) \leq Q_{loss}^{\max} \tag{14c}$$

$$\sum_{r'} f_t(r'|r) = 1,\ \forall r, \text{and } f_t(r'|r) \geq 0,\ \forall r, r'. \tag{14d}$$

Inequalities (14b) are the series of linear constraints (13), repeated for each value of r'; inequality (14c) reflects the service quality constraint; constraints (14d) reflect that $f_t(r'|r)$ is a probability distribution function.

4.2. Optimal Strategy for the Adversary

To find the adversary strategy that is at a game theoretical equilibrium with the user strategy computed in (14a-14d) (i.e., the two strategies are optimal against each other), we construct and solve the linear program that is dual to (14a-14d): Choose $h_t(\hat{r}|r'), y_r, \forall r, r', \hat{r}$, and $z \in [0, \infty)$ to

$$\textbf{Minimize} \ \sum_{r} \psi_t(r)\, y_r + z Q_{loss}^{\max} \tag{15a}$$

$$\textbf{subject to} \ y_r \geq \sum_{\hat{r}} h_t(\hat{r}|r') d_p(\hat{r}, r) - z d_q(r', r), \forall r, r' \tag{15b}$$

$$z \geq 0 \tag{15c}$$

$$\sum_{\hat{r}} h_t(\hat{r}|r') = 1, \forall r', \text{and } h_t(\hat{r}|r') \geq 0, \forall r', \hat{r}. \tag{15d}$$

Note the role of variable z: In linear programming parlance, it is the *shadow price* of the service quality constraint. Intuitively, z is the "exchange rate" between service quality and privacy. Its value in the optimal solution indicates the amount of privacy (in privacy units) that is lost (gained) if the service quality threshold $Q_{loss}^{\max}$ increases (decreases) by one unit of quality.

For example, if $z > 0$ in the optimal solution, then any change $\Delta Q_{loss}^{\max}$ in $Q_{loss}^{\max}$ will change the privacy achieved by $z \Delta Q_{loss}^{\max}$. In this case, constraint (14c) is satisfied as a strict equality. In contrast, if constraint (14c) is satisfied as a strict inequality, then, intuitively, the selection of $f_t(r'|r)$ has not been constrained by $Q_{loss}^{\max}$. In this case, any (small) changes in $Q_{loss}^{\max}$ will have no effect on $f_t(r'|r)$, nor on the privacy achieved. So, z would be zero.

Note that both linear programs compute the unconditional expected privacy of the user (3), which we repeat here for convenience.

$$Privacy(\psi_t, f_t, h_t, d_p) = \sum_{\hat{r}, r', r} \psi_t(r) f_t(r'|r) h_t(\hat{r}|r') d_p(\hat{r}, r). \tag{16}$$

The optimal solution of each linear program results in the same value for the privacy of the user. Hence, in principle, we only need to compute one of the two to quantify the maximum level of privacy of the user. We choose to present both, because the user's linear program incorporates the service quality constraint in a more straightforward manner, whereas the adversary's linear program explicitly computes the "exchange rate" between service quality and privacy.

5. LPPM DESIGN IN SPECIFIC SCENARIOS

We now formalize concrete problems, starting with sporadic location exposures and continuing with the problems that were described in Section 2.2.

Compared to our description so far, the variations in the following problems have to do with the previously observed pseudolocations (which affect the attacker's prior knowledge $\psi_t()$ about his objective) and with the attacker's objective (which may be not just be to estimate the single current location r_t, but also past locations r_{t-1} or future locations r_{t+1}, so this also affects the computation of $\psi_t()$). In other words, the central quantity that changes among the various problems is the adversary's prior knowledge about his objective. For each particular problem, we show how to compute $\psi_t()$, which can then be simply inserted into the linear programs (14) and (15). Finally, we show how one can adapt the framework for arbitrary new problems.

Problem 0: Protecting privacy of time t at time t, with no previous exposure

The user has previously not exposed any pseudolocation. Currently, at time t, the user issues her first query, and the LPPM must choose an appropriate obfuscation function that generates r'_t to protect the user's current location r_t.

This problem is equivalent to that of sporadic location privacy, which is relevant for LBSs to which queries are sent *sporadically*: location check-in, location-tagging, or applications for finding nearby points-of-interest, local events, or nearby friends. In these services, if there is enough time between queries, then successively exposed locations are independent of each other, i.e. previous exposures do not give any information to the attacker about the current exposure.

The relevant prior information in this case is $\Pr\{r_t\}$ and it is computed from background knowledge about the area in which the user moves (e.g. if r is in the middle of a lake, then presumably $\Pr\{r_t\} = 0$), and also from background knowledge about the user's mobility or habits (e.g. if the attacker knows the user's daily mobility pattern, then at t = 7am $\Pr\{r_t\}$ is high for locations around the user's home, and at 11am it is high around the user's workplace). If the mobility is described by a Markov chain, then $\Pr\{r_t\}$ can be simply read from the steady state distribution of the Markov chain.

Problem 1: Protecting privacy of time t at time t, for a given exposure at $t-1$

At time $t-1$, the LPPM exposed pseudolocation r'_{t-1}. Currently, at time t, the user issues another query, so the LPPM must choose an appropriate obfuscation r'_t to protect the user's current location r_t. This is the problem shown in Figure 2 (Section 2.2).

The attacker, having observed r'_{t-1}, has a probabilistic estimate of the user's location at time $t-1$, which was previously (at $t-1$) computed as $h_{t-1}(\hat{r}_{t-1}|r'_{t-1})$. This estimate must be “moved forward” to form a location estimate for time t. If we assume that the user moves according to a Markov chain mobility model, “moving forward” from $t-1$ to t is equivalent to multiplying $h_{t-1}(\hat{r}_{t-1}|r'_{t-1})$ by the Markov chain transition matrix $\Pr\{r_t|r_{t-1}\}$. Note that the Markov chain assumption is not necessary for our proposed approach; all we need is a way to compute a location estimate at time t from a location estimate at time $t-1$. We make the Markov assumption for convenience; other literature has used simpler models, based on just the velocity of the user [Ghinita et al. 2009].

$$\Pr\{r_t|r'_{t-1}\} = \sum_{r_{t-1}} \Pr\{r_t, r_{t-1}|r'_{t-1}\} = \sum_{r_{t-1}} \Pr\{r_t|r_{t-1}\} h_{t-1}(\hat{r}_{t-1}|r'_{t-1}) \tag{17}$$

Function $\Pr\{r_t|r'_{t-1}\}, r_t \in \mathcal{R}$ is the prior information of the adversary for Problem 1. The LPPM design continues with solving the linear program as in the previous section, computing $f(r'_t|r_t, r'_{t-1})$ for a particular value of r'_{t-1}.

Note, however, that the prior information function $\Pr\{r_t|r'_{t-1}\}$, and thus the designed LPPM, is different for each value of the previously exposed r'_{t-1}. This is a formal description of the obvious fact that the LPPM to be designed must depend on the exact value of the previously exposed pseudolocation.

This dependence raises the practically important issue of when to compute the appropriate LPPM. One choice is to precompute all LPPMs, one for each possible value of r'_{t-1} (M in total), and then copy them to the device. This allows for performing all computations on a powerful, non-resource-constrained machine, but it requires more storage space on the mobile device. The other extreme is to compute the appropriate LPPM when the actual r'_{t-1} becomes known, but this means that the computation has to happen on the mobile device between time $t-1$ and time t. So, there is a tradeoff here, which we explore when discussing computational considerations (Section 6.2).

Of course there are intermediate solutions, such as precomputing and storing LPPMs only for the most frequent pseudolocation values. The probability of each pseudolocation can be computed as

$$\Pr\{r'\} = \sum_{r} \Pr\{r\} \Pr\{r'|r\} = \sum_{r} \Pr\{r\} f(r'|r). \tag{18}$$

A variant of Problem 1 is when the latest exposure was at time $t-k$, for a single value of $k \geq 1$. Then we compute the prior by multiplying $h_{t-k}(\hat{r}_{t-k}|r'_{t-k})$ by the k-th power of the Markov transition matrix. Note here that, if k is large enough, then the effect of $h_{t-k}(\hat{r}_{t-k}|r'_{t-k})$ disappears, and so we are back to the original sporadic problem of protecting privacy at time t with just $\Pr\{r_t\}$ as the prior, as if there are no prior exposures. We explore this variant next, where our general aim is to quantify the effect of multiple past exposures, as opposed to just one.

Problem 1A: Exposing past pseudolocations and its effect on the adversary’s knowledge of the user’s current location

In a continuous LBS, users expose more than one pseudolocation in short order. The question is to what extent pseudolocations exposed before $t-1$ give information to the adversary about the user’s current (time t) location. If pseudolocations far into the past need to be taken into account, then an LPPM that protects the current location would be correspondingly more complex, as each possible combination of past exposures would induce a different optimal current obfuscation; in other words, a different LPPM would have to be computed for each combination.

To answer this question, we compare the sequence of priors $\Pr\{r_t\}$, $\Pr\{r_t|r'_{t-1}\}$, $\Pr\{r_t|r'_{t-2}\}, \cdots$. The general term of this sequence, $\Pr\{r_t|r'_{t-k}\}$ is equal to

$$\Pr\{r_t|r'_{t-k}\} = \sum_{r_{t-k}} \Pr\{r_t, r_{t-k}|r'_{t-k}\} = \sum_{r_{t-k}} \Pr\{r_t|r_{t-k}\} h_{t-k}(r_{t-k}|r'_{t-k}). \tag{19}$$

The term $\Pr\{r_t|r_{t-k}\}$ is the k-step transition probability in a Markov chain mobility model; it is computed as the k-th power of the corresponding transition matrix.

So we see that the effect of r'_{t-k} on the adversary's prior on the current location comprises two factors: (1) the uncertainty for the true location r_{t-k} at time $t-k$, which is caused by the obfuscation used at time $t-k$, and (2) the dissipation effect caused by the k-step transition from $t-k$ to t.

Problem 2: Protecting privacy of times $t-1$ and t at time t, for a given exposure at $t-1$

At time $t-1$, the LPPM exposed pseudolocation r'_{t-1}. Currently, at time t, the user issues another query, so the LPPM must choose an appropriate obfuscation r'_t that will protect the current location r_t and will not retroactively compromise the user's *previous* location r_{t-1}. In other words, the obfuscation r'_t must be "compatible" with the previously exposed r'_{t-1}. This is the problem shown in Figure 3 (Section 2.2).

The prior information that is available to the adversary before he observes r'_t is

$$\Pr\{r_t, r_{t-1}|r'_{t-1}\} = \Pr\{r_t|r_{t-1}, r'_{t-1}\} \Pr\{r_{t-1}|r'_{t-1}\} = \Pr\{r_t|r_{t-1}\} h_{t-1}(r_{t-1}|r'_{t-1}). \tag{20}$$

The first term $\Pr\{r_t|r_{t-1}\}$ is known from the mobility model (e.g. the Markov transition matrix), and the second term $h_{t-1}(r_{t-1}|r'_{t-1})$ is computed at $t-1$.

Note that here the prior information is not just about the current location r_t, but rather about the pair (r_t, r_{t-1}), because that pair is the information that the adversary tries to infer.

Problem 3: Protecting privacy of times $t+1$ and t at time t

In this Problem, we assume that there are no past exposures of the user's location. The user issues a query at the current time t, and her objective is to protect not only the current location, but also the location at the next time instant $t+1$.

We discussed this scenario at the end of Section 2.2, and it is motivated as follows: Disclosing the current location might not be important in and of itself, but it might make it much easier for the adversary to infer the next location, which happens to be very sensitive. For instance, the user might currently be on a street that only leads to an abortion clinic. Hence, disclosing her current location is almost equivalent to disclosing that she will go to the clinic. Symmetrically, her current location might be very sensitive, and her next (expected) location can be linked easily to her current one. For instance, she might be about to leave the abortion clinic and enter a street that is only used as the clinic's exit. Furthermore, neither the current nor the next location might be particularly sensitive separately, but the transition from one to the other might be.

The conclusion in all these cases is that the current location must be protected jointly with the (possible) next one(s) where the user will be at time $t+1$. For this reason, the LPPM should compute at the present time t the pseudolocations it is likely to output at $t+1$, so that the current choice of r'_t does not limit future choices. The intuition is that the LPPM should choose the current pseudolocation r'_t so that future paths that the user will likely take can be protected with pseudolocations compatible with r'_t.

In this scenario, the prior information of the adversary is about the current (t) and future ($t+1$) location of the adversary: $\Pr\{r_{t+1}, r_t\} = \Pr\{r_{t+1}|r_t\} \Pr\{r_t\}$. The first term is known from the Markov transition matrix, and the second term is, as in Problem 0, just the steady state of the Markov chain.

As noted earlier, in all four problems we can compute an appropriate prior probability distribution for the adversary's knowledge about the target locations. This probability distribution is then used as a parameter in the linear program described in the previous section. In Section 5.1 below, we see how one can write a linear program for a very general set of target locations and prior exposures.

When problems are interleaved

As the user moves, she might be facing a different problem at each time, so she would need to compute a succession of different LPPMs. For example, the first time she exposes a location, Problem 0 applies. Then, at the next exposure, Problem 1 or Problem 2 could apply, depending on whether the user wishes to protect just the current location (Problem 1) or both the current and the previously exposed location (Problem 2). The prior probability at each time has to be computed taking into account the past LPPMs and, most importantly, their corresponding attacks h: It is these attacks that determine the attacker's probabilistic estimate of the user's location at each time, which is then multiplied by the transition probabilities to provide the prior probability for the next time instant.

We now compute the prior probability at time t, $t \geq 2$, in an example where the user originally computed a Problem-0 LPPM at time $t = 1$, and then at every subsequent time instant up to $t - 1$ she computed a Problem-2 LPPM. She is currently (time t) interested in computing yet another Problem-2 LPPM, hence the prior probability is $\Pr\{r_t, r_{t-1} | r'_{t-1}\}$. We denote by h^{spor} the Problem-0 attack.

The computation of the prior proceeds as follows:

$$\Pr\{r_t, r_{t-1} | r'_{t-1}\} = \Pr\{r_t | r_{t-1}, r'_{t-1}\} \Pr\{r_{t-1} | r'_{t-1}\}. \tag{21}$$

But the first term is known from the user's mobility model: $\Pr\{r_t | r_{t-1}, r'_{t-1}\} = \Pr\{r_t | r_{t-1}\}$. If $t = 2$, the second term $\Pr\{r_{t-1} | r'_{t-1}\}$ can be immediately equated to the attack h at time $t - 1$: $\Pr\{r_{t-1} | r'_{t-1}\} = h^{\text{spor}}_{t-1}(r_{t-1} | r'_{t-1})$, as above. So this concludes the case $t = 2$.

To compute $\Pr\{r_{t-1} | r'_{t-1}\}$ for the case $t > 2$, we use Bayes' rule:

$$\Pr\{r_{t-1} | r'_{t-1}\} = \frac{\Pr\{r'_{t-1} | r_{t-1}\} \Pr\{r_{t-1}\}}{\sum_{r_{t-1}} \Pr\{r'_{t-1} | r_{t-1}\} \Pr\{r_{t-1}\}} \tag{22}$$

Now, $\Pr\{r_{t-1}\}$ is known (again from the mobility model; if the mobility is a Markov chain, then this is the steady state of the Markov chain), so we only need to compute $\Pr\{r'_{t-1} | r_{t-1}\}$:

$$\begin{aligned}
\Pr\{r'_{t-1} | r_{t-1}\} &= \sum_{r'_{t-2}, r_{t-2}} \Pr\{r'_{t-1}, r_{t-2}, r'_{t-2} | r_{t-1}\} \\
&= \sum_{r'_{t-2}, r_{t-2}} \Pr\{r'_{t-1} | r_{t-1}, r_{t-2}, r'_{t-2}\} \Pr\{r_{t-2}, r'_{t-2} | r_{t-1}\} \\
&= \sum_{r'_{t-2}, r_{t-2}} \Pr\{r'_{t-1} | r_{t-1}, r_{t-2}, r'_{t-2}\} \Pr\{r'_{t-2} | r_{t-2}, r_{t-1}\} \Pr\{r_{t-2} | r_{t-1}\} \quad (23)
\end{aligned}$$

The first term is the LPPM function f_{t-1} as computed at time $t-1$ for Problem 2. The third term is known from the mobility model. The second term $\Pr\{r'_{t-2} | r_{t-2}, r_{t-1}\}$ is equal to $\Pr\{r'_{t-2} | r_{t-2}\}$, because the obfuscation at time $t - 2$ depends only on r_{t-2}, r_{t-3},

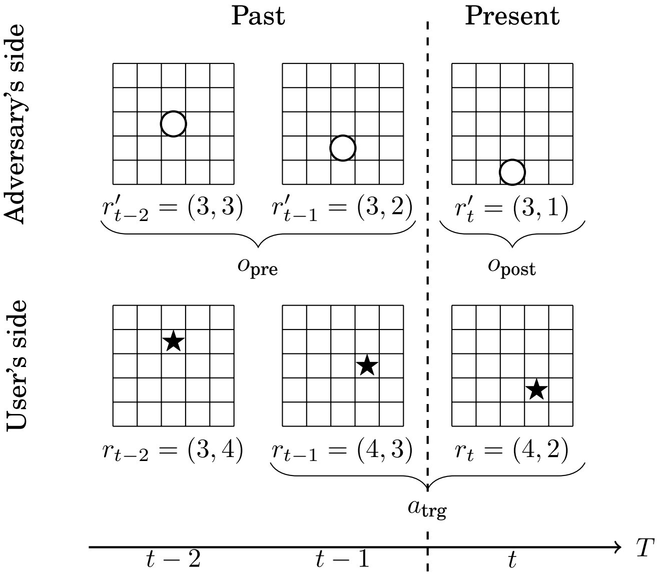


Fig. 4. A user moves from location $(3,4)$ at time $t-2$, to (4,3) at $t-1$, to $(4,2)$ at current time t. The user wants to protect locations at times $t-1$ and t, and these are denoted by a_{trg} (target events). At past times $t-2$ and $t-1$, the LPPM exposed pseudolocations $(3,3)$ and $(3,2)$ (denoted by o_{pre}), and to protect a_{trg} the LPPM currently exposes location $(3,1)$ (denoted by o_{post}).

and r'_{t-3}. Knowing r_{t-1} when r_{t-2} is already known gives us no extra information on r_{t-3} or r'_{t-3}. Hence, the computation of $\Pr\{r'_{t-1}|r_{t-1}\}$ is shown to be recursive:

$$\Pr\{r'_{t-1}|r_{t-1}\} = \sum_{r'_{t-2},r_{t-2}} f_{t-1}(r'_{t-1}|r_{t-1},r_{t-2},r'_{t-2})\Pr\{r'_{t-2}|r_{t-2}\}\Pr\{r_{t-2}|r_{t-1}\}. \quad (24)$$

Having recursively computed $\Pr\{r'_{t-1}|r_{t-1}\}$, we substitute it into (22), the result of which is in turn substituted into (21) to compute the desired prior.

5.1. Location privacy for a generic objective

As we see in the previous sections, many different variants of location privacy can be formulated, depending on the adversary's knowledge (i.e., past exposed pseudolocations), and on the privacy target (i.e., on what the user wishes to protect or, equivalently, on what the adversary wishes to attack). Each combination leads to a different optimal LPPM. In particular, in Problems 2 and 3, we see that the privacy target does not need to be the user's current location. It can be a pair or a tuple of locations, and this tuple might not even include the user's current location, e.g., if the user only wants to protect a past location by choosing an appropriate pseudolocation at the current time.

We now describe generic terminology and a generic linear program for LPPM design, with the help of the example shown in Figure 4.

5.1.1. Generic LPPM parameters

— a_{trg} denotes the target events that the user wants to protect, or equivalently, the events that the adversary wants to infer. In the example, the user wants to protect her location at times $t-1$ and t and thus $a_{\text{trg}} = (r_{t-1}, r_t) = \{\langle(4,3), t-1\rangle, \langle(4,2), t\rangle\}$.

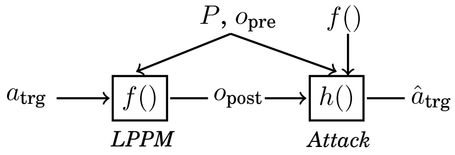


Fig. 5. Information available to the LPPM and the adversary: The LPPM wants to protect location(s) a_{trg} by producing appropriate pseudolocation(s) o_{post}. The adversary observes the output o_{post} of the LPPM and, using his knowledge of the LPPM function f, estimates a_{trg}; the adversary's estimate is $\hat{a}_{\text{trg}}$. The prior knowledge of the adversary and of the LPPM consists of the transition matrix P and the pseudolocations o_{pre} that have been produced in the past.

- o_{pre} is a subset of the pseudoevents that the LPPM created and sent to the LBS *up to but before* the current time. These are the pseudoevents that matter for the estimation of a_{trg}: Typically, o_{pre} would be a sequence of consecutive pseudoevents starting with a recent time instant (as old ones do not matter for estimating a_{trg}) and leading up to the current time. These are known both to the adversary and to the LPPM. In the example, the relevant pseudolocations were exposed at times $t-1$ and $t-2$ and thus $o_{\text{pre}} = (r'_{t-2}, r'_{t-1}) = \{\langle(3,3), t-2\rangle, \langle(3,2), t-1\rangle\}$.
- o_{post} is the pseudolocation (or set of pseudolocations) that the LPPM produces to protect a_{trg} and that will be sent to the LBS at the current time. In the example, at current time t the user exposes pseudolocation $(3,1)$ thus $o_{\text{post}} = (r'_t) = \{\langle(3,1), t\rangle\}$.
- $f(o_{\text{post}}|a_{\text{trg}}, o_{\text{pre}})$ is the probability that the LPPM produces o_{post}, given its knowledge o_{pre} and the locations a_{trg} it is trying to protect. This function encodes the defensive mechanism. It can be viewed as a codebook that prescribes, for each value of a_{trg} and o_{pre}, a randomization over the possible values of o_{post}.

5.1.2. Generic privacy metric. As before, privacy is quantified as the adversary's error in estimating the user's true location(s) a_{trg}. Figure 5 illustrates the information flow of events and pseudoevents to the LPPM and to the adversary. The detailed notation is as follows:

- $\psi(a_{\text{trg}}|o_{\text{pre}})$ is the adversary's prior probability distribution on the inference target a_{trg}, given his prior knowledge o_{pre}. It encodes what the adversary can deduce about a_{trg} *before* observing the LPPM's current output o_{post}.
- $\hat{a}_{\text{trg}}$ denotes the adversary's estimate of a_{trg}. Similarly to a_{trg}, it can be seen as a time-indexed vector whose elements belong to the set $\mathcal{R}$ of locations.
- $h(\hat{a}_{\text{trg}}|o_{\text{pre}}, o_{\text{post}})$ is the probability that the adversary estimates $\hat{a}_{\text{trg}}$ to be the true value of a_{trg}, given his knowledge of prior pseudolocations o_{pre} and given the pseudolocation(s) o_{post} exposed at current time t.
- $d_p(\hat{a}_{\text{trg}}, a_{\text{trg}}) \geq 0$ is the privacy gain when the adversary's estimate is $\hat{a}_{\text{trg}}$ and the true value of the inference target is a_{trg}. It is zero only if $\hat{a}_{\text{trg}} = a_{\text{trg}}$.

The privacy that an LPPM $f(.)$ achieves against an adversary implementing attack $h(.)$ is then the expected value of $d_p(\hat{a}_{\text{trg}}, a_{\text{trg}})$, given prior observations o_{pre}:

$$
\begin{aligned}
Privacy(\psi, f, h, d_p; o_{\text{pre}}) &= \mathbb{E}\{d_p(\hat{a}_{\text{trg}}, a_{\text{trg}})|o_{\text{pre}}\} = \sum_{a_{\text{trg}}, \hat{a}_{\text{trg}}} \mathbf{Pr}\{\hat{a}_{\text{trg}}, a_{\text{trg}}|o_{\text{pre}}\} d_p(\hat{a}_{\text{trg}}, a_{\text{trg}}) \\
&= \sum_{\substack{a_{\text{trg}} \\ o_{\text{post}} \\ \hat{a}_{\text{trg}}}} \psi(a_{\text{trg}}|o_{\text{pre}}) f(o_{\text{post}}|a_{\text{trg}}, o_{\text{pre}}) h(\hat{a}_{\text{trg}}|o_{\text{pre}}, o_{\text{post}}) d_p(\hat{a}_{\text{trg}}, a_{\text{trg}}).
\end{aligned}
\tag{25}
$$

This formula represents the adversary's expected estimation error.

5.1.3. Generic quality metric. The final ingredient is the quality metric:

— q_{trg} denotes the relevant events with respect to quality. Similarly to a_{trg}, q_{trg} is a time-indexed vector. However, its time indices are not necessarily the same as those of a_{trg}: The locations/times that matter for quality may be different from the ones that matter for privacy.

— $d_q(q_{\text{trg}}, o_{\text{post}}, o_{\text{pre}})$ represents the quality loss when q_{trg} is the true value of the quality-relevant events, the LPPM currently reports o_{post} and it has reported o_{pre} in the past.

The expected quality loss caused by an LPPM $f(.)$ is the expected value of $d_q(q_{\text{trg}}, o_{\text{post}}, o_{\text{pre}})$ over all q_{trg} and o_{post}, for a given history o_{pre}.

5.1.4. Generic linear program. We now form the linear program that computes the optimal LPPM (using, as before, auxiliary variables $x_{o_{\text{post}}}$):

We want to maximize $\sum_{o_{\text{post}}} x_{o_{\text{post}}}$ under the constraint

$$x_{o_{\text{post}}} \leq \sum_{a_{\text{trg}}} \psi(a_{\text{trg}}|o_{\text{pre}}) f(o_{\text{post}}|a_{\text{trg}}, o_{\text{pre}}) d_p(\hat{a}_{\text{trg}}, a_{\text{trg}}), \forall \hat{a}_{\text{trg}}, o_{\text{post}}, \tag{26}$$

under the quality constraint,

$$\sum_{\substack{q_{\text{trg}} \\ o_{\text{post}}}} \Pr\{q_{\text{trg}}|o_{\text{pre}}\} \Pr\{o_{\text{post}}|q_{\text{trg}}, o_{\text{pre}}\} d_q(q_{\text{trg}}, o_{\text{post}}, o_{\text{pre}}) \leq Q_{loss}^{\max}, \tag{27}$$

as well as the constraint (omitted) that f should be a probability function.

6. EVALUATION

The LPPMs that we design are optimal by construction. In this section, we illustrate their privacy-utility performance, and we also compare them against non-optimal LPPMs. We also show how the optimal attacks that we compute fare against non-optimal, but intuitive, LPPMs. We begin with LPPMs and attacks computed for the sporadic case (Problem 0), and we continue with trajectory-aware LPPMs.

6.1. Location Obfuscation for Sporadic Exposures

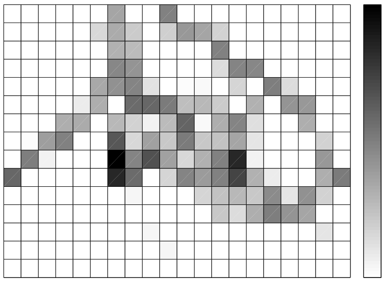


Fig. 6. Spatial histogram showing the density of users per region (in log scale) in Lausanne. Size is $15.32\text{km} \times 7.58\text{km}$, divided into 20×15 regions.

We use real location traces of people (in Lausanne, Switzerland) who use various means of transportation.[2] We select 11 users at random, and we focus on their location traces during the day (8am to 8pm), when it is more probable that users use location-based services. The length of the considered traces is one month. The location area, within which they move, is divided into 300 regions. Figure 6 shows the density of users across all the regions. The grayness of the cells shows the density of its corresponding region in log scale. As many of the regions are abandoned (or very rarely visited) by many individual users, we compute each user's profile $\psi(.)$ by considering only the 30 most popular regions across the whole population. This prevents sparse user profiles. A user's profile is the normalized number of her visits to each region.

[2] The traces are obtained from the Nokia Lausanne Data Collection Campaign dataset.

Given distance functions $d_p(.)$ and $d_q(.)$ and service-quality loss threshold $Q_{loss}^{\max}$, we compute the optimal LPPM and its corresponding optimal attack by solving (14a) and (15a) using Matlab's linear programming solver. We then compare the obtained optimal protection mechanism and the optimal inference attack against the *basic obfuscation LPPM* and the *Bayesian inference attack*, respectively.

Basic Obfuscation LPPM. The basic obfuscation LPPM, with an obfuscation level $k = 1, 2, 3, \ldots$, is constructed in the following way: For each location r, we find its $k-1$ closest locations (using the Squared Euclidean distance between the centers of the regions). The LPPM function $f(.|r)$ will be the uniform probability distribution on the set of the $k-1$ selected locations together with the location r. That is, location r is replaced by each of the k locations, as a pseudolocation, with the same probability $\frac{1}{k}$, and all other locations have probability 0. Thus, in practice, an actual location r is hidden among its $k-1$ nearest locations. We choose this mechanism, as it has been very popular in the literature.

Given the user profile $\psi(.)$ and quality distance function $d_q(.)$, we use (1) to compute the expected service-quality loss $Q_{loss}(\psi, f, d_q)$ for any LPPM obfuscation $f(.)$, whether it be optimal or not.

Bayesian Inference Attack on an LPPM. We compare the effectiveness of our optimal attack with the Bayesian inference attack, which has been shown effective before [Shokri et al. 2011b]. The Bayesian attack first computes the posterior probability of locations $\Pr(.|r')$:

$$\Pr(r|r') = \frac{\Pr(r, r')}{\Pr(r')} = \frac{f(r'|r)\psi(r)}{\sum_r f(r'|r)\psi(r)}, \forall r \in \mathcal{R}. \tag{28}$$

Then, it sets $\hat{r}$ to the location that minimizes the expected estimation error $\hat{r} = \arg\min_{\hat{r}} \sum_r \Pr(r|r') d_p(\hat{r}, r)$. If the estimation error d_p is the Squared Euclidean distance, then the attack selects the conditional expected value $\hat{r} = \mathrm{E}[r|r']$ and the resulting expected estimation error is the conditional variance $\mathrm{Var}[r|r']$. If d_p is the Discrete Metric, then the attack selects the location with the highest posterior probability $\hat{r} = \arg\max_{\hat{r}} \Pr(\hat{r}|r')$ and the resulting expected estimation error is $1 - \sum_{r \neq \hat{r}} \Pr(r|r')$. These facts follow from standard Bayesian estimation theory.

The difference between the Bayesian attack and the optimal attack is that the Bayesian attack does not take into account that the LPPM was designed to anticipate the optimal attack against it, nor that the LPPM had a quality constraint. In contrast, both of these facts are incorporated into the linear program that computes the optimal attack. As a result, the Bayesian attack is not optimal against any particular LPPM, definitely not against the optimal defense LPPM that our algorithms design.

Optimal Inference Attack on an Arbitrary LPPM. In order to make a fair comparison between the effectiveness of the optimal and obfuscation LPPM, we need to run the same attack on both of them. The Bayesian inference attack described by (28) can be performed against both. However, we still need to design an optimal attack against arbitrary LPPMs that have not been constructed in our game-theoretic framework.

The optimal inference attack is the one that minimizes the expected user privacy:

$$h(.) = \arg\min_h Privacy(\psi, f, h, d_p). \tag{29}$$

Given the user profile $\psi(.)$, an LPPM $f(.)$ and distortion function $d_p(.)$, the following linear program finds the optimal attack $h(.)$. Note that, compared to (15a), there is no

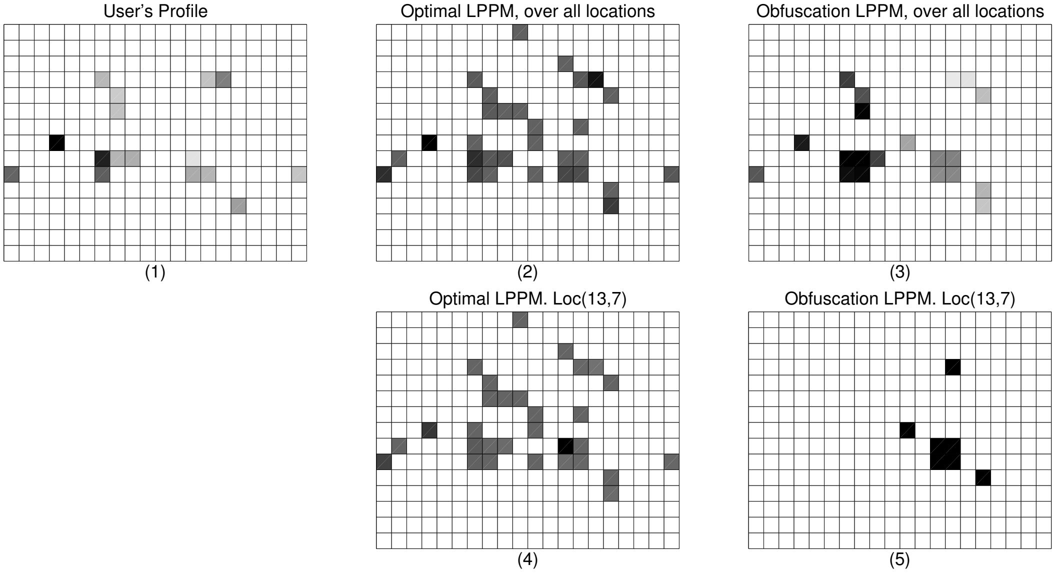


Fig. 7. Input/Output of LPPM. Profile of a user for whom the subsequent calculations are made (sub-figure 1). Distribution $\Pr(r')$ of observed pseudolocations when using the optimal LPPM with $Q_{loss}^{\max} = 0.8690$ (sub-figure 2). Distribution $\Pr(r')$ of observed pseudolocations when using obfuscation LPPM with $Q_{loss}(\psi, f, d_q) = 0.8690$ (sub-figure 3). Conditional distribution $\Pr(r'|r)$ when using the optimal LPPM on location $r = (13, 7)$ (sub-figure 4). Conditional distribution $\Pr(r'|r)$ when using obfuscation LPPM on location $r = (13, 7)$ (sub-figure 5). Column 1 is the leftmost column, and row 1 is the bottom row. (Squared Euclidean d_p, Discrete Metric d_q)

service quality constraint here, as the LPPM has been assumed to be arbitrary.

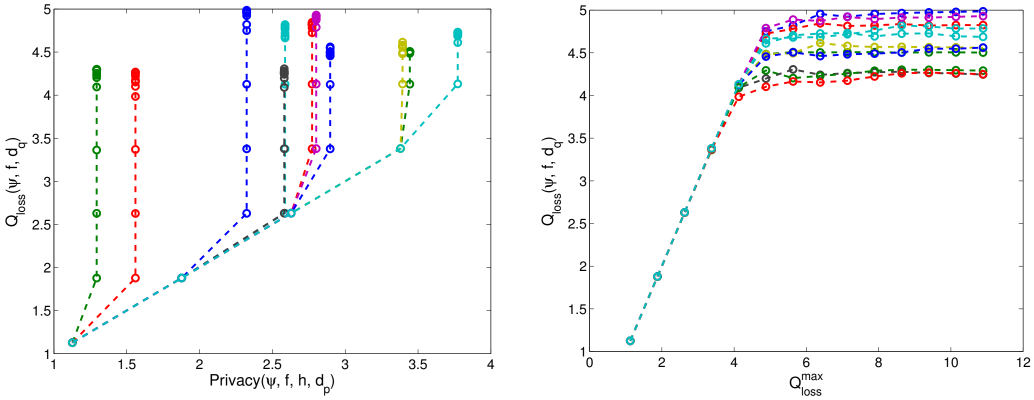
$$\textbf{Minimize} \sum_{\hat{r},r',r} \psi(r) f(r'|r) h(\hat{r}|r') d_p(\hat{r}, r) \tag{30a}$$

$$\textbf{subject to} \sum_{\hat{r}} h(\hat{r}|r') = 1, \forall r', \text{and } h(\hat{r}|r') \geq 0, \forall \hat{r}, r' \tag{30b}$$

Tradeoff between Privacy and Service Quality. We now study the tradeoff between the level of privacy that the optimal LPPM provides, against the optimal attack, and the service-quality loss that it causes. We plot in Figure 8(a) the evolution of the service quality loss and the corresponding privacy achieved, as the optimal LPPM is configured with higher and higher values of service quality thresholds $Q_{loss}^{\max}$ (for users with diverse profiles). Each line in the figure represents one user and each ○ represents one $Q_{loss}^{\max}$. We plot $Privacy(\psi, f, h, d_p)$ versus $Q_{loss}(\psi, f, d_q)$.

Unsurprisingly, with higher levels of location-privacy protection comes a significant degradation in service quality. Also, as expected, the maximum achievable location privacy strongly depends on the user profile. This is reflected by the separation between the different lines. But we also see that each user's privacy increases up to a certain level and then there is no change even when the quality threshold $Q_{loss}^{\max}$ is further increased. This is due to the presence of the optimal attack that squeezes the location-privacy gain.

This effect is further illustrated in Figure 8(b), where the service-quality loss of the optimal LPPM is plotted against the service-quality threshold. Once the optimal LPPM offers the maximal location privacy for a given user profile, loosening the service-



(a) Location privacy $Privacy(\psi, f, h, d_p)$ vs. Service-quality loss $Q_{loss}(\psi, f, d_q)$ for a given service-quality threshold $Q^{\max}_{loss}$. The circles ∘ represent different values of $Q^{\max}_{loss}$.

(b) Service-quality threshold $Q^{\max}_{loss}$ vs. Service-quality loss $Q_{loss}(\psi, f, d_q)$, for a given level of location privacy $Privacy(\psi, f, h, d_p)$. The circles ∘ represent different values of $Privacy(\psi, f, h, d_p)$.

Fig. 8. Tradeoff between Privacy and Service Quality: Optimal LPPM against the optimal attack. The different lines represent users with diverse profiles $\psi(.)$. (Squared Euclidean $d_q(.)$ and Squared Euclidean $d_p(.)$.)

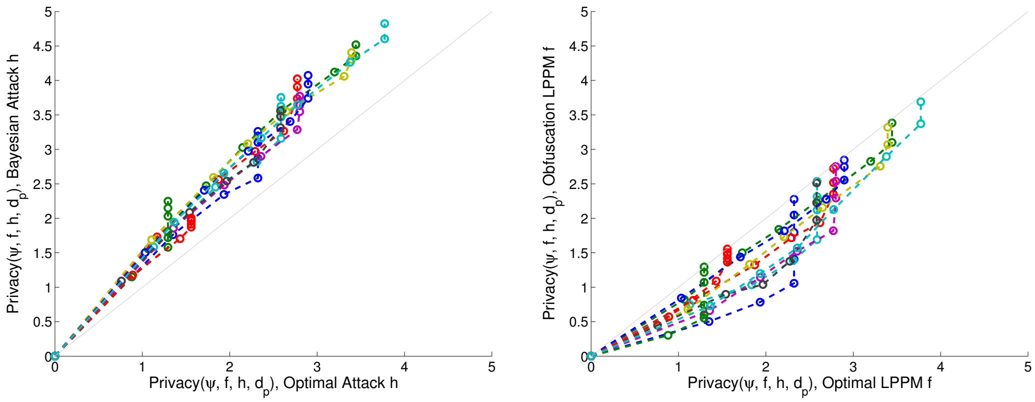
quality constraint does not significantly change the LPPM's underlying function f, and thus there is no reduction in service quality. In other words, there is no need to sacrifice any more service quality, even though the looser $Q^{\max}_{loss}$ constraint allows it, because doing so does not increase the user's location privacy.

We can draw some parallels with the "shadow price" interpretation of z in the linear program (Section 4.2): In that section, we see that $z = 0$ means that small changes to $Q^{\max}_{loss}$ have no effect on privacy, whereas a positive value of z gives the rate of privacy increase for a small change in $Q^{\max}_{loss}$. Qualitatively, we observe both these effects in the figure (an initial privacy increase with $Q^{\max}_{loss}$, subsequently no effect on privacy as $Q^{\max}_{loss}$ increases further). However, note that the "shadow price" interpretation is only valid for small changes of $Q^{\max}_{loss}$, so, strictly speaking, we cannot quantitatively link the values of z to the observed privacy changes in the figure.

Optimal LPPM and attack are better than Basic LPPM and Bayesian attack. Given Squared Euclidean distance functions $d_p(.)$ and $d_q(.)$, we compute the optimal LPPM and attack methods for a set of service quality thresholds $Q^{\max}_{loss}$. For each user, we run the Bayesian inference attack on her optimal LPPM. We also evaluate the location privacy offered by the basic obfuscation LPPM with respect to the optimal attack. We vary the obfuscation level from 1 (minimum) to 30 (maximum), and for each level we compute the corresponding quality loss. Then, this value is set as the threshold $Q^{\max}_{loss}$ in the computation of the optimal attack mechanism.

Figure 9(a) shows the superiority of the optimal attack to the Bayesian attack, when location privacy of users is protected using the optimal LPPM: For any given user and service-quality threshold, the location privacy that the user obtains is smaller when the adversary implements the optimal strategy rather than the Bayesian attack.

Figure 9(b) shows the superiority of the optimal LPPM to the obfuscation LPPM, against the optimal attack: For any given user and service-quality threshold, a user has a higher privacy level when the LPPM implements the optimal strategy. Note that the privacy levels achieved by the two mechanisms approach each other and eventually become equal when very little service quality is guaranteed for the user (i.e., when



(a) Location privacy $Privacy(\psi, f, h, d_p)$ offered by the optimal LPPM against the optimal attack derived using the game theoretic approach vs. against the Bayesian-inference attack.

(b) Location privacy $Privacy(\psi, f, h, d_p)$ offered by the optimal LPPM vs. location privacy offered by the basic obfuscation LPPM, both evaluated against the optimal attack.

Fig. 9. Effectiveness of the optimal attack and optimal LPPM strategies. Lines represent users with different profiles $\psi(.)$. Circles $\circ$ represent different values of $Q_{loss}^{\max}$, each of which corresponds to a different obfuscation level of the basic obfuscation LPPM. (Squared Euclidean $d_q(.)$ and Squared Euclidean $d_p(.)$.)

$Q_{loss}^{\max}$ is set to its maximum value). When the quality requirement becomes looser and looser, both mechanisms add so much noise that the adversary in effect learns nothing new about the user's true location.

6.2. Location Obfuscation over a Trajectory

So far, we analyzed the behavior of optimal LPPMs in the sporadic setting. In this section, we focus on LPPMs that take the previous exposed locations into account. We also evaluate transition privacy.

For the comparison to the sporadic LPPM and for the illustration of the privacy-quality tradeoff, we use a real data set of location traces. These traces, which are one day long, belong to 10 randomly chosen mobile users (vehicles) in the San Francisco Bay area from the epfl/mobility dataset at CRAWDAD. We divide the Bay Area into 10×25 equal-size locations, and consider a day to be composed by 288 time units, one per each 5 minutes. We emphasize that the granularity of both time and locations can be arbitrarily selected depending on the required accuracy in quantifying privacy and service quality. We consider all the locations that are visited by each user, which on average is 23.4 locations per user. We also consider all the transitions that each user has made between these locations in our dataset.

Without loss of generality, we select the privacy gain d_p and the quality loss d_q functions to be the Discrete Metric: $d_p(\hat{a}_{trg}, a_{trg}) = 1_{\hat{a}_{trg} \neq a_{trg}}$ and $d_q(q_{\text{trg}}, o_{\text{post}}, o_{\text{pre}}) = 1_{q_{\text{trg}} \neq (o_{\text{post}}, o_{\text{pre}})}$. Taking the privacy gain as an example, using the Discrete Metric implies that we only consider it bad for privacy when the attacker correctly estimates the *exact* value of the target locations (i.e., when $\hat{a}_{trg}$ is exactly equal to a_{trg}). All other estimates are equally good for privacy, regardless, e.g., of the physical distance between the attacker's estimate and the true value of a_{trg}. As our quantification of privacy is the *expected* value of $d_p(\hat{a}_{trg}, a_{trg})$ – and the expected value of $1_{\hat{a}_{trg} \neq a_{trg}}$ is just the probability of $\hat{a}_{trg} \neq a_{trg}$, which is the adversary's probability of error.

For the maximum tolerable quality loss $Q_{loss}^{\max}$, we do not specify a single value, but rather compute the achievable privacy for multiple values, so as to observe the privacy-quality tradeoff.

For the previously reported events o_{pre}, we do not specify a single value. Instead, the privacy values that we compute are averaged over all possible values of o_{pre}, because such an average is more representative of the privacy that a user can expect to achieve:

$$\sum_{o_{\text{pre}}} \Pr\{o_{\text{pre}}\} Privacy(\psi, f, h, d_p; o_{\text{pre}}).$$

Comparison to Optimal Sporadic LPPM. A trajectory-oblivious (sporadic) LPPM is typically evaluated against an attack that is also sporadic, i.e., an attack in which location correlation is not taken into account. To provide quantitative justification for the inadequacy of such LPPMs and their evaluation when the exposed locations are correlated, we show in Figure 10 that a correlation-aware attack can achieve much lower privacy than a sporadic attack.

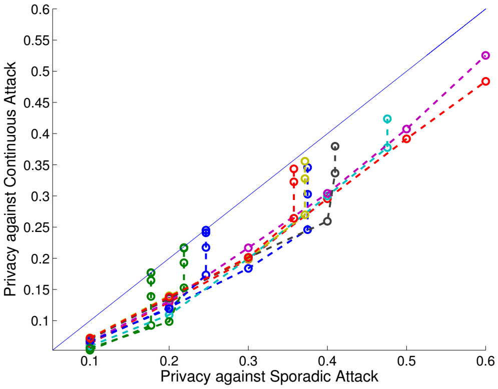


Fig. 10. Users' single-location privacy, using a sporadic LPPM against two attacks: sporadic attack vs. correlation-aware attack. For 10 different users (lines), and for various values of the service quality threshold $Q_{loss}^{\max}$ (dots), we see that the privacy against a correlation-aware attack (x-axis) is always less than the privacy against a sporadic attack (y-axis).

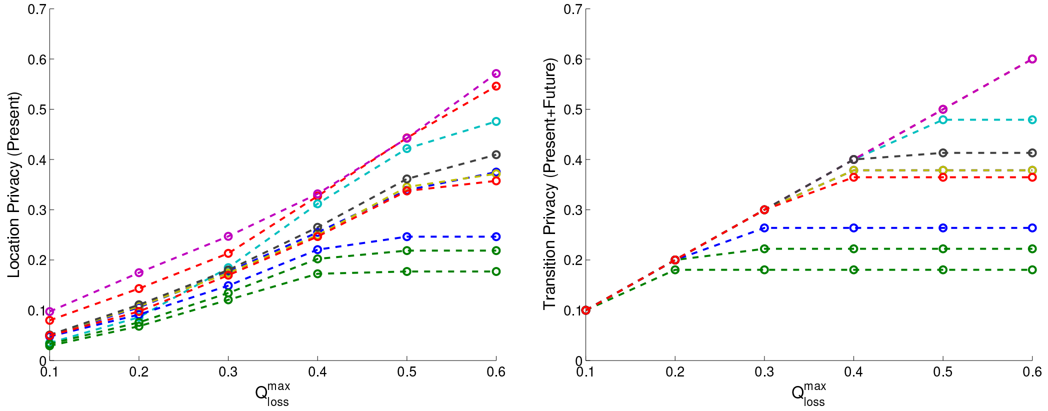
A sporadic LPPM protects single locations only, so to compare meaningfully, we pick as objective of the correlation-aware attack the single-location privacy objective, i.e., $a_{\text{trg}} = r_t$. The difference between the correlation-aware attack and the sporadic attack is that the former uses the conditional prior probability on the target location $\psi(r_t|o_{\text{pre}})$ (for $o_{\text{pre}} = o_{t-1}$), whereas the latter uses the unconditional prior $\psi(r_t)$.

Each attack is paired against the same sporadic LPPM, and the results are plotted across the 10 mobile users and for various values of the service quality threshold $Q_{loss}^{\max}$. As all data points are below the $x = y$ diagonal, we conclude that privacy in the correlation-aware attack (x-axis) is lower than privacy in the sporadic attack (y-axis). The only cases where the two attacks are equally (un-)successful are when the quality loss threshold is so high that the sporadic LPPM can inject enough noise to blur even the inference of a correlation-aware attack.

Privacy-Quality Tradeoff. Here, we illustrate the privacy-quality tradeoff of our LPPMs for two particular scenarios: Protecting single-location privacy for the current location, taking into account the immediately previous pseudolocation ($a_{\text{trg}} = r_t$ and $o_{\text{pre}} = o_{t-1}$), shown in Figure 11(a); and protecting transition privacy for the current and future locations ($a_{\text{trg}} = (r_t, r_{t+1})$), shown in Figure 11(b).

Under each of these two scenarios, we construct the optimal protection mechanism for each of the 10 users in our traces (i.e., the mechanism that provides the maximum privacy for her). We plot this maximum privacy as a function of the service quality threshold $Q_{loss}^{\max}$. We see in both figures that the achievable privacy increases as $Q_{loss}^{\max}$ increases (as higher values of $Q_{loss}^{\max}$ let the LPPM inject more noise).

Similarly to the sporadic case, we observe two effects: First, a saturation effect takes place for most users as $Q_{loss}^{\max}$ increases. Their privacy reaches a plateau beyond which any further increase in $Q_{loss}^{\max}$ does not increase privacy. Second, the privacy plateau, as well as the privacy level for any value of $Q_{loss}^{\max}$, differs for each user, indicating that there is an inherent per-user privacy limit that is connected to how predictable the user's mobility is.



(a) Single-location privacy: Protecting the current location ($a_{\text{trg}} = r_t$), when o_{pre} is the pseudolocation reported in the previous time instant o_{t-1}.

(b) Transition privacy: Protecting the current and the next location ($a_{\text{trg}} = (r_t, r_{t+1})$).

Fig. 11. Tradeoff between Privacy and Service Quality. Each curve corresponds to one user.

Computational Considerations. Our mechanism is intended to be computed offline and used online: The LPPM function $f(o_{\text{post}}|a_{\text{trg}}, o_{\text{pre}})$ is precomputed offline and then downloaded to the device. Then, whenever the user attempts to expose a location, the LPPM looks up and performs the appropriate randomization on pseudolocations o_{post}, based on the actual values of the target events to be protected a_{trg} and the previously exposed vector of pseudolocations o_{pre}. In this way, the only computational burden of the mobile device is a look-up and a randomized selection of o_{post}.

The offline computation of the LPPM function f requires solving a separate linear program for each value of o_{pre} that may arise in practice. But most of the theoretically possible values of the vector o_{pre} are nonsensical sequences of locations, e.g., sequences where successive locations are too far away from each other, so these need not be taken into account, which saves considerable time. Similarly, the number of variables in each linear program is theoretically equal to the total number of pairs of a_{trg} and o_{post} vectors, since a value for f must be computed for each such combination. This number is $M^{\text{length}(a_{\text{trg}})+\text{length}(o_{\text{post}})}$ (recall that M is the total number of locations), but in practice it is much smaller. The actual number of linear programs and of variables is closer to the number of likely trajectories of the corresponding length (the number of linear programs is equal to the number of trajectories of length length(o_{pre}), whereas the number of variables is equal to the number of trajectories of length length$(a_{\text{trg}}) +$ length(o_{post})).

It is very important to notice also that the computation of f needs to be done only once, so the associated cost only needs to be incurred once. A recomputation of f is only necessary if, for example, the user parameters or application parameters d_p, d_q, Q^{max}_{loss} change, or if the user wants to protect a different aspect of her privacy (e.g., previous, present, and next location, instead of just present and next location), which would translate to a change in a_{trg}, or if one wishes to take into account different prior knowledge of previously reported pseudolocations o_{pre} (e.g., take into account the 3 previously reported pseudolocations instead of just one).

7. RELATED WORK

Location privacy has been a very active area of research in recent years. Work on this topic can be roughly classified in three categories: mainly focused on the design of LPPMs; mainly focused on recovering actual user trajectories from anonymized or perturbed traces; or mainly focused on the formal analysis and the search for appropriate location privacy metrics to allow for fair comparison between LPPMs.

Existing LPPMs are built according to different design principles. The most popular approach to obtaining location privacy is to send a space- or time-obfuscated version of the users' actual locations to the service provider [Gedik and Liu 2005; Gruteser and Grunwald 2003; Hoh et al. 2007; Kalnis et al. 2007]. A different approach consists in hiding some of the users' locations by using mix zones [Beresford and Stajano 2003; Freudiger et al. 2009], or silent periods [Jiang et al. 2007]. These are regions where users do not communicate with the provider while changing their pseudonym. Provided that several users traverse the zone simultaneously, this mechanism prevents an adversary from tracking them, as he cannot link those who enter with those who exit the region. A third line of work protects location privacy by adding dummy requests, indistinguishable from real requests, issued from fake locations to the service provider [Chow and Golle 2009]. The purpose of these fake locations is to increase the uncertainty of the adversary about the users' real movements.

The predictability of users' location traces, and the particular constraints of users' movements, has been shown to be sufficient to reconstruct and/or identify anonymous or perturbed locations. An adversary can, to name but a few possibilities, infer users' activities from the frequency of their visits to certain locations [Liao et al. 2007]; re-identify anonymous low-granularity location traces given the users' mobility profiles [De Mulder et al. 2008]; or derive [Hoh et al. 2006], and re-identify [Golle and Partridge 2009; Krumm 2007] the home address of individuals from location traces.

Several authors have made efforts towards formalizing the desirable location privacy requirements that LPPMs should fulfill, as well as towards finding suitable metrics to evaluate the degree to which these requirements are fulfilled. Examples of these lines of work are Krumm [2007], Decker [2009], and Duckham [2010]. Shokri *et al.* [2009] revisit existing LPPMs and the location-privacy metrics used in their evaluation. They classify these metrics in three categories: uncertainty-based (entropy), error-based and k-anonymity. The authors conclude, by means of a qualitative evaluation, that metrics such as entropy and k-anonymity are not suitable for measuring location privacy. In a follow-up of this work, Shokri *et al.* [2011a; 2011b] provide a framework to quantify location privacy. The framework allows us to specify an LPPM and then to evaluate various questions about the location information leaked. Our design methodology uses this analytical framework as an evaluation tool to quantifying the LPPMs' offered privacy against the localization attack.

Specifically for protecting trajectory privacy (i.e. consecutive location exposures), a first class of mechanisms in the literature protect user privacy when trajectories are published in bulk. Protection is achieved by grouping trajectories of different users in a wide area to ensure that the aggregate trajectory can be ascribed to at least k users [Abul et al. 2008]; mixing the trajectories of k users [Nergiz et al. 2009]; eliminating some events from the published dataset [Hoh et al. 2010; Terrovitis and Mamoulis 2008]; or replacing locations with larger regions defined by a pre-defined grid [Gidófalvi et al. 2007]. Along similar lines, some protection algorithms need access to the complete trajectory before protection can be applied [You et al. 2007], or they delay the exposure of queries so as to gather additional information about subsequent user locations [Ghinita et al. 2009; Ardagna et al. 2012]. In contrast, our approach decides in real time how to protect a given location that the user is about to expose.

Other trajectory-aware mechanisms assume the existence of a trusted third party (e.g. the cellular service provider) [Pan et al. 2009; Gao et al. 2013], or assume that nearby users are present and can be leveraged to achieve joint privacy protection [Beresford and Stajano 2003; Freudiger et al. 2009; Huang et al. 2006]. Both of these scenarios violate the user-centricity design requirement in this paper. Not depending on other users is also the reason why k-anonymity does not apply in our case, as well as any other method that attempts to make users indistinguishable.

Despite the extent to which location privacy has been studied, there is a patent disconnection between these different lines of work. Most of the aforementioned papers use different models to state the problem and evaluate location privacy. This hinders the comparison of systems and slows down the design of robust LPPMs. Further, in some of these papers there is a detachment between the proposed design and the adversarial model against which it is evaluated. Often the considered adversary is static in its knowledge and disregards the information leaked by the LPPM algorithm; or adversarial knowledge is not even considered in the evaluation. The works by Freudiger *et al.* [2009] and Shokri *et al.* [2009; 2011a; 2011b] do consider a strategic adversary that exploits the information leaked by the LPPM in order to compute location privacy. Nevertheless, their work does not address how this privacy computation can be integrated in the design of location-privacy preserving mechanisms.

In this work, we bridge the gap between design and evaluation of LPPMs. We provide a systematic method for developing LPPMs; our method maximizes users' location privacy while guaranteeing a desired level of service quality. We formalize the optimal design problem as a Bayesian Stackelberg game similar to previous work on security in which, as in our location-privacy scenario, the defender can be modeled as a Stackelberg game leader, and the adversary as the follower. The common theme with this previous work is that the defender must commit to a defense strategy/protocol, which is then disclosed to the adversary, who can then choose an optimal course of action *after* observing the defender's strategy. Paruchuri *et al.* [2008] propose an efficient algorithm for finding the leader's optimal strategy considering as a main case study a patrolling agent who searches for a robber in a limited area. In their case, the defender is unsure about the type of the adversary (i.e. where the adversary will attack). In contrast, in our work it is the adversary who is unsure about the type (i.e. the true location) of the user/defender. A similar approach is used by Liu and Chawla [2009] in the design of an optimal e-mail spam filter, taking into account that spammers adapt their e-mails to get past the spam detectors. The same problem is tackled by Brückner and Scheffer [2011], who further compare the Stackelberg-based approach with previous spam filters based on support vector machines, logistic regression, and Nash-logistic regression. Korzhyk *et al.* [2011] contrast the Stackelberg framework with the more traditional Nash framework, within a class of security games. A recent survey [Manshaei et al. 2013] explores the connections between security and game theory more generally. To the best of our knowledge, our work is the first that uses Bayesian Stackelberg games to design optimal privacy-protection mechanisms.

The only other formal approach to measuring and protecting privacy that we are aware of is by Andrés *et al.* [2013], who extend the concept of differential privacy to sporadic location privacy, thus defining a new privacy metric: geo-indistinguishability. They also propose a mechanism to achieve it optimally [Bordenabe et al. 2014]. In later work [Chatzikokolakis et al. 2014], geo-indistinguishability is extended to correlated location traces. To achieve this extended notion of geo-indistinguishability, the concrete mechanism proposed reports the previous pseudolocation (i.e. $r'_t = r'_{t-1}$) if that is acceptable in terms of quality (i.e. if r'_{t-1} is close enough to r_t), or adds zero-mean Laplacian noise to the true location r_t otherwise. The main feature of geo-indistinguishability in both the sporadic and in the correlated setting is that it does not use any information about the adversary's prior knowledge, whereas our approach does. We consider this to be a design choice for the LPPM designer, rather than an objective advantage or disadvantage of one method over the other, as it in effect models a different adversary. If the LPPM designer wants to protect against an adversary with some background knowledge, then our approach is the only one possible. Otherwise, one can use either our approach with an uninformative prior (i.e. with a uniform prior over all possible locations), or the no-prior approach. The technical difference between the two is that our approach aims to maximize the Bayesian estimation error, whereas no-prior approaches aim to keep the likelihoods of nearby locations close to each other. In a recent work, Shokri [2015] shows how to combine the two notions of differential and distortion privacy, and how to optimize their joint effect on privacy and utility. However, it does not address trajectories and correlated locations.

8. CONCLUSION

Accessing location-based services from mobile devices entails a privacy risk for users, since sensitive information can be inferred from the locations they visit. This information leakage raises the need for robust location-privacy protecting mechanisms (LPPMs). In this paper, we have proposed a game-theoretic framework that enables a designer to find the optimal LPPM for a given location-based service, ensuring a satisfactory service quality for the user. This LPPM is designed to provide user-centric location privacy, hence it is ideal to be implemented in mobile devices.

Our method accounts for the fact that the strongest adversary not only observes the perturbed location sent by the user but also knows the algorithm implemented by the protection mechanism. Hence, he can exploit the information leaked by the LPPM's algorithm to reduce his uncertainty about the user's true location. In our approach, the user is aware of the adversary's knowledge and does not make any assumption about attacker's computation power. Hence, she prepares the protection mechanism against the strongest possible attack by modeling the problem as a Bayesian Stackelberg competition.

We have validated our method using real location traces. We have demonstrated that our approach finds the optimal attack for a given LPPM and service-quality constraint, and we have shown that it is superior to other LPPMs such as basic location obfuscation. We have also shown that the superiority of the optimal LPPM over alternatives is more significant when the service-quality constraint imposed by the user is tightened. Hence, our solution is effective exactly where it will be used. Finally, our results confirm that loosening the service-quality constraint allows for increased privacy protection, but the magnitude of this increase strongly depends on the user profile, i.e., on the degree to which a user's location is predictable from her LBS access profile. To the best of our knowledge, this is the first framework to explicitly include the adversarial knowledge into a privacy-preserving design process, considering the *common knowledge* between the privacy protector and the attacker.

REFERENCES

Osman Abul, Francesco Bonchi, and Mirco Nanni. 2008. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. In *24th International Conference on Data Engineering (ICDE 2008)*. IEEE, 376–385.

Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential Privacy for Location-based Systems. In *ACM Conference on Computer and Communications Security (CCS'13)*. ACM, 901–914.

Claudio A Ardagna, Giovanni Livraga, and Pierangela Samarati. 2012. Protecting privacy of user information in continuous location-based services. In *15th International Conference on Computational Science and Engineering (CSE)*. IEEE, 162–169.

Alastair R. Beresford and Frank Stajano. 2003. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing* 2, 1 (2003), 46–55. DOI:http://dx.doi.org/10.1109/MPRV.2003.1186725

Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2014. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 251–262.

Michael Brückner and Tobias Scheffer. 2011. Stackelberg games for adversarial prediction problems. In *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, Chid Apté, Joydeep Ghosh, and Padhraic Smyth (Eds.). ACM, 547–555.

Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. 2014. A predictive differentially-private mechanism for mobility traces. In *Privacy Enhancing Technologies*. Springer, 21–41.

Richard Chow and Philippe Golle. 2009. Faking contextual data for fun, profit, and privacy. In *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*. ACM, New York, NY, USA, 105–108. DOI:http://dx.doi.org/10.1145/1655188.1655204

Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. 2008. *Algorithms*. McGraw-Hill, NY.

Yoni De Mulder, George Danezis, Lejla Batina, and Bart Preneel. 2008. Identification via location-profiling in GSM networks. In *WPES '08: Proceedings of the 7th ACM workshop on Privacy in the electronic society*. ACM, New York, NY, USA, 23–32. DOI:http://dx.doi.org/10.1145/1456403.1456409

Michael Decker. 2009. Location Privacy - An Overview. In *International Conference on Mobile Business*. IEEE Computer Society, 221–230.

Matt Duckham. 2010. Moving forward: location privacy and location awareness. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS (SPRINGL '10)*. ACM, New York, NY, USA, 1–3. DOI:http://dx.doi.org/10.1145/1868470.1868472

Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. 2009. On the Optimal Placement of Mix Zones. In *PETS '09: Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*. Springer-Verlag, Berlin, Heidelberg, 216–234. DOI:http://dx.doi.org/10.1007/978-3-642-03168-7_13

Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. 2012. Evaluating the privacy risk of location-based services. In *Proceedings of the 15th international conference on Financial Cryptography and Data Security (FC'11)*. Springer-Verlag, Berlin, Heidelberg, 31–46. DOI:http://dx.doi.org/10.1007/978-3-642-27576-0_3

Sheng Gao, Jianfeng Ma, Weisong Shi, Guoxing Zhan, and Cong Sun. 2013. TrPF: A Trajectory Privacy-Preserving Framework for Participatory Sensing. *IEEE Transactions on Information Forensics and Security* 8, 6 (June 2013), 874–887. DOI:http://dx.doi.org/10.1109/TIFS.2013.2252618

Bugra Gedik and Ling Liu. 2005. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*. IEEE Computer Society, Washington, DC, USA, 620–629.

Gabriel Ghinita, Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. 2009. Preventing velocity-based linkage attacks in location-aware applications. In *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (ACM-GIS 2009)*. ACM, 246–255.

Győző Gidófalvi, Xuegang Huang, and Torben Bach Pedersen. 2007. Privacy-Preserving Data Mining on Moving Object Trajectories. In *8th International Conference on Mobile Data Management (MDM 2007)*. IEEE, 60–68.

Philippe Golle and Kurt Partridge. 2009. On the Anonymity of Home/Work Location Pairs. In *Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing*. Springer-Verlag, Berlin, Heidelberg, 390–397. DOI:http://dx.doi.org/10.1007/978-3-642-01516-8_26

Marco Gruteser and Dirk Grunwald. 2003. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, New York, NY, USA, 31–42. DOI:http://dx.doi.org/10.1145/1066116.1189037

Michael Herrmann, Carmela Troncoso, Claudia Díaz, and Bart Preneel. 2013. Optimal sporadic location privacy preserving systems in presence of bandwidth constraints. In *12th annual ACM Workshop on Privacy in the Electronic Society*, Ahmad-Reza Sadeghi and Sara Foresti (Eds.). ACM, 167–178.

Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. 2006. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE Pervasive Computing* 5, 4 (2006), 38–46.

Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. 2007. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. ACM, New York, NY, USA, 161–171.

Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. 2010. Achieving Guaranteed Anonymity in GPS Traces via Uncertainty-Aware Path Cloaking. *IEEE Transactions in Mobile Computing* 9, 8 (2010), 1089–1107.

Leping Huang, Hiroshi Yamane, Kanta Matsuura, and Kaoru Sezaki. 2006. Silent cascade: Enhancing location privacy without communication QoS degradation. In *Security of Pervasive Computing (SPC)*. 165–180.

Tao Jiang, Helen J. Wang, and Yih-Chun Hu. 2007. Preserving location privacy in wireless LANs. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*. ACM, New York, NY, USA, 246–257. DOI:http://dx.doi.org/10.1145/1247660.1247689

P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. 2007. Preventing Location-Based Identity Inference in Anonymous Spatial Queries. *Knowledge and Data Engineering, IEEE Transactions on* 19, 12 (Dec. 2007), 1719–1733. DOI:http://dx.doi.org/10.1109/TKDE.2007.190662

D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. 2011. Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness. *Journal of Artificial Intelligence Research* 41 (May–August 2011), 297–327.

John Krumm. 2007. Inference Attacks on Location Tracks. In *In Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive), volume 4480 of LNCS*. Springer-Verlag, 127–143.

Lin Liao, Donald J. Patterson, Dieter Fox, and Henry A. Kautz. 2007. Learning and inferring transportation routines. *Artif. Intell.* 171, 5-6 (2007), 311–331.

Wei Liu and Sanjay Chawla. 2009. A Game Theoretical Model for Adversarial Learning. In *IEEE International Conference on Data Mining Workshops (ICDM 2009)*, Yücel Saygin, Jeffrey Xu Yu, Hillol Kargupta, Wei Wang, Sanjay Ranka, Philip S. Yu, and Xindong Wu (Eds.). IEEE Computer Society, 25–30.

Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Başar, and Jean-Pierre Hubaux. 2013. Game Theory Meets Network Security and Privacy. *ACM Comput. Surv.* 45, 3, Article 25 (July 2013), 39 pages. DOI:http://dx.doi.org/10.1145/2480741.2480742

Joseph Meyerowitz and Romit Roy Choudhury. 2009. Hiding stars with fireworks: location privacy through camouflage. In *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, New York, NY, USA, 345–356.

Kristopher Micinski, Philip Phelps, and Jeffrey S Foster. 2013. An Empirical Study of Location Truncation on Android. *MoST'13: Proc. of the Mobile Security Technologies* 2 (2013).

Mehmet Ercan Nergiz, Maurizio Atzori, Yücel Saygin, and Baris Güç. 2009. Towards Trajectory Anonymization: a Generalization-Based Approach. *Transactions on Data Privacy* 2, 1 (2009), 47–75.

Xiao Pan, Xiaofeng Meng, and Jianliang Xu. 2009. Distortion-based anonymity for continuous queries in location-based mobile services. In *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (ACM-GIS 2009)*. 256–265.

Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. 2008. Efficient Algorithms to Solve Bayesian Stackelberg Games for Security Applications. In *23rd AAAI Conference on Artificial Intelligence (AAAI 2008)*, Dieter Fox and Carla P. Gomes (Eds.). AAAI Press, 1559–1562.

Reza Shokri. 2015. Privacy Games: Optimal User-Centric Data Obfuscation. *Proceedings of Privacy Enhancing Technologies* (2015).

Reza Shokri, Julien Freudiger, Murtuza Jadliwala, and Jean-Pierre Hubaux. 2009. A distortion-based metric for location privacy. In *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*. ACM, New York, NY, USA, 21–30. DOI:http://dx.doi.org/10.1145/1655188.1655192

Reza Shokri, George Theodorakopoulos, George Danezis, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2011a. Quantifying location privacy: the case of sporadic location exposure. In *Proceedings of the 11th international conference on Privacy enhancing technologies (PETS'11)*. Springer-Verlag, Berlin, Heidelberg, 57–76. http://dl.acm.org/citation.cfm?id=2032162.2032166

Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. 2011b. Quantifying Location Privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy (SP '11)*. IEEE Computer Society, Washington, DC, USA, 247–262. DOI:http://dx.doi.org/10.1109/SP.2011.18

Manolis Terrovitis and Nikos Mamoulis. 2008. Privacy Preservation in the Publication of Trajectories. In *9th International Conference on Mobile Data Management (MDM '08)*. IEEE, 65–72.

Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. 2007. Protecting Moving Trajectories with Dummies. In *Mobile Data Management, 2007 International Conference on*. 278–282.

Received July 2015; revised June 2016; accepted October 2016